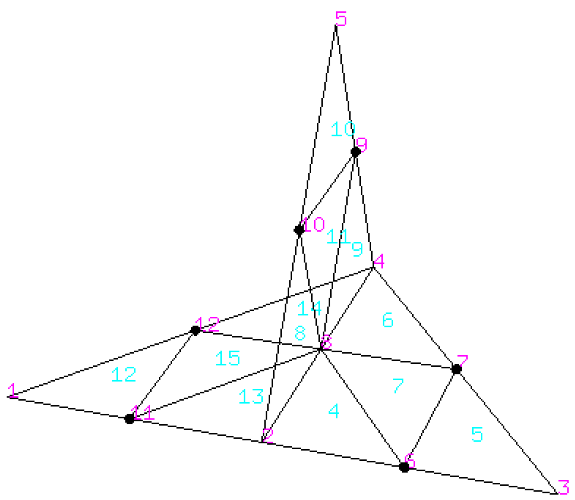
European Commission

JRC TECHNICAL REPORTS

Adaptivity in Shell/Beam/Bar Elements in EUROPLEXUS

Folco Casadei
Martin Larcher
George Valsamos

2014



B-hanging nodes after SPLI 1

EUROPLEXUS Software

Report EUR 26697 EN

Joint Research Centre

European Commission
Joint Research Centre
Institute for the Protection and Security of the Citizen

Contact information
Martin Larcher
Address: Joint Research Centre, Via Enrico Fermi 2749, TP 480, 21027 Ispra (VA), Italy
E-mail: martin.larcher@jrc.ec.europa.eu
Tel.: +39 0332 78 9563
Fax: +39 0332 78 9049

http://ipsc.jrc.ec.europa.eu/
http://www.jrc.ec.europa.eu/

This publication is a Technical Report by the Joint Research Centre of the European Commission.

JRC90456

EUR 26697 EN

ISBN 978-92-79-38670-1
ISSN 1831-9424
doi:10.2788/85346

Luxembourg: Publications Office of the European Union, 2014

Printed in Italy

Contents

List of Figures

List of Tables

1. Introduction

This report is a sequel to reports and publications [1-14] on mesh adaptivity in fast transient dynamics and presents the implementation of mesh adaptivity in shell, beam and bar elements in fast transient dynamics. The algorithms are implemented in the EUROPLEXUS code. In particular, a 4-node shell element in 3D (Q4GS), a 3-node shell element in 3D (T3GS) and 2-node elements capable of representing shells in 2D or beams/bars either in 2D or in 3D are considered (ED01).

EUROPLEXUS [15] is a computer code for fast explicit transient dynamic analysis of fluid-structure systems jointly developed by the French Commissariat à l'Energie Atomique et aux Energies Alternatives (CEA Saclay) and by the Joint Research Centre of the European Commission (JRC Ispra).

Reference [1] presented the first implementation in EUROPLEXUS of an adaptive mesh refinement and un-refinement procedure, in two space dimensions (element shape QUA4) for solid mechanics. The procedure was extended to fluid mechanics (FE formulation) in 2D in reference [2]. Then, reference [3] applied a similar refinement and un-refinement procedure in three space dimensions to the CUB8 element shape, both in solids mechanics and in fluid mechanics (FE formulation).

All numerical examples presented in references [1-3] with a variable mesh used a so-called "manual" mesh adaptation directive, the `WAVE` directive (see the code manual in reference [15]), first introduced in reference [1]. This directive refines the mesh along "wavefronts" that are specified by the user, e.g. according to a known analytical solution to the problem considered. This technique was used with success to simulate a bar problem (in solid mechanics) and a shock tube problem (in fluid mechanics) both in 2D and in 3D [1-3].

However, those solutions cannot be qualified as "true" adaptive solutions, because in (true) adaptivity mesh refinement and un-refinement should be completely automatic, based upon suitable *error estimators* or *error indicators*. The formulation of error estimators in fast transient dynamics is challenging and is still a subject of research. The use of so-called error indicators, however, is much simpler. For this reason, subsequent work in EUROPLEXUS focused on error indicators. References [4] and [5] document a first prototype implementation of adaptivity based upon error indicators in EUROPLEXUS, limited to 2D problems in continuum and fluid mechanics. An extension of the indicator technique to 3D is under development but has not been completed and documented yet.

Publications [6-7] focus on the natural quantities of interest in goal-oriented error assessment and adaptivity, but limited to the case of linear elasto-dynamics.

The adaptive technique was then applied to Cell-Centred Finite Volumes (CCFV) for the description of the fluid domain, first in 2D (see [8]) and then also in 3D [9]. More recently, the technique has also been extended for use with the CDEM combustion model which makes use of the CCFV formulation [10]. A complete description of the element refinement and un-refinement techniques used in mesh adaptation has been published in a paper [11]. Finally, reference [12] shows the combination of mesh adaptivity with Fluid-Structure Interaction, i.e. the automatic fluid mesh refinement and un-refinement near a moving and deforming structure.

Reference [13] extends mesh adaptivity to simplex element shapes, i.e. the 3-node triangle (TRI3) in 2D and the 4-node tetrahedron (TET4) in 3D. These elements are useful in fully general unstructured meshing of complex geometries. The extension covered solid continuum elements (CEA's TRIA and TETR elements *with solid material*), and for fluid Finite Elements (JRC's FL23 and FL34 elements) and Cell-Centered Finite Volumes (CEA's T3VF and TEVF elements).

Reference [14] shows the extension of adaptivity to some of CEA's continuum elements (CAR1, TRIA, CUBE, TETR) when used with an ALE or Eulerian formulation for the treatment of fluids. The numerical fluxes are generalized to the adaptive case, similarly to what is done also for JRC's specialized fluid Finite Elements.

The present work is a first attempt at extending mesh adaptivity to 3D shell elements. The Q4GS 4-node shell is taken as a first example because it is one of the most efficient and used shell elements in EPX. Then, also the T3GS triangle (which is similar to Q4GS as far as formulation is concerned) is considered. Finally, the case of 2D shells, i.e. of 2-node "segment-like" elements, is considered (this element shape can also represent beams or bars in either 2D or 3D). The treatment of shells/beams/bars presents some particularities with respect to the continuum elements considered in previous work, and these are highlighted below.

This document is organized as follows:

- Section 2 presents the treatment of the Q4GS shell element. The treatment of T3GS is similar in many aspects (only the number of nodes, the number of Gauss points and the number of edges are different) and is briefly mentioned at the end of the Section. Finally, 2-node segment-like elements (e.g. ED01) are also considered shortly.
- Section 3 presents some numerical examples.
- The references are listed in Section 4.

The Appendix contains a listing of all the input files mentioned in the present report.

2. Mesh adaptation

To begin, the splitting and unsplitting of a quadrilateral shell element in 3D (Q4GS) is described. Overall, the same strategy as reference [1] for the 4-node continuum quadrilateral in 2D is adopted, see e.g. Figure 1, leading to the appearance of so-called "hanging nodes" in zones of the mesh where the refinement is not uniform.

However, with respect to the 2D continuum case there are some important differences. First of all, we are dealing here with three spatial dimensions instead of just two. But more importantly, shell elements usually have six degrees of freedom per node, three translations (like in the continuum case) and three rotations. For the element chosen, rotations are expressed in the global reference frame, and this facilitates the treatment with respect to other shell elements in EPX whose rotations are expressed in local (node-based) reference systems. So the continuity of both translations *and rotations* will have to be imposed at hanging nodes.

Another aspect that distinguishes shells from continuum elements is the possibility of having not only neighbor elements, but also "adjacent" elements (such terms are better defined below), and in general any number of these can be present since shell structures often present junctions.

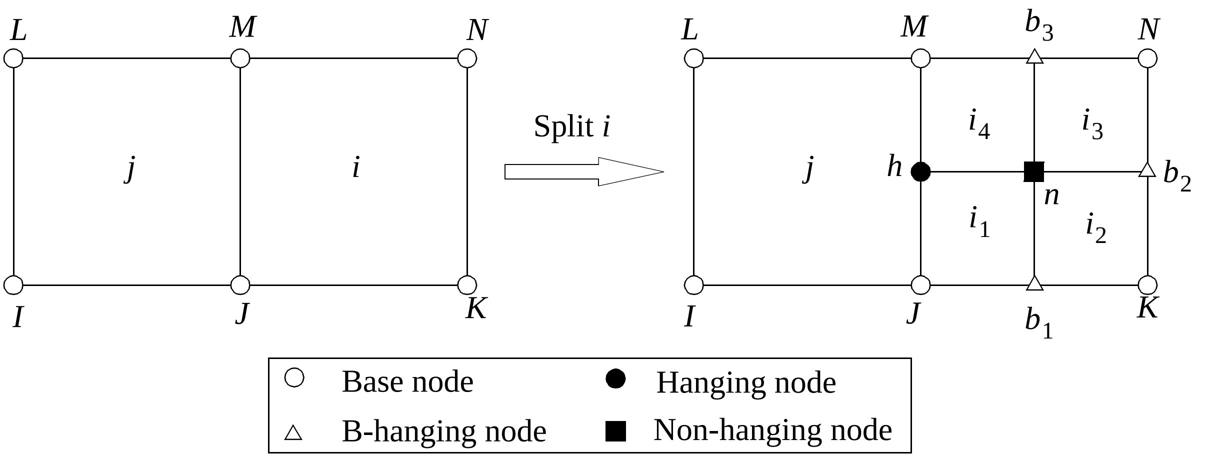


Figure 1 - Splitting a Q4GS

Consider the simple mesh of Figure 1, containing just two Q4GS elements i and j, sharing a common edge. The splitting of element i is shown in the right part of the Figure. Like in the continuum case, (up to) five new nodes are generated. The central node n is always generated: this node is independent of any others and its degrees of freedom (dofs) are "true" new dofs which are added to the model. Along the four edges, new nodes may or may not be created depending on the presence or not of adjacent elements and on the level of these elements in the adaptivity tree. To illustrate these concepts we have to set some terminology first.

Shell element faces

A (3D) shell element has two *faces*, an “upper” face and a “lower” face (by convention). For example, element i of Figure 1 has face $IJML$ as upper face (the oriented normal points “upwards” i.e. outside the page in this case) and face $ILMJ$ as lower face. Only the sense of numbering matters, not the initial node of the face (I in this case), because this defines the oriented normal.

The distinction between upper and lower face is therefore a bit subtle. Algorithms that would depend upon the orientation of the face (distinction between upper and lower) can be quite delicate.

A shell element may have other elements “attached” to its faces. The most common case is that of so-called “boundary condition” elements (CLxx), used in EPX to imposed natural conditions such as for example an applied pressure. In analogy to the continuum case (reference [1]) these elements are called *neighbors*, see Figure 2 for a simple example and the precise definition given below. Although the shell has two faces, usually only (at most) one neighbor can be present, again due to the fact that the distinction between upper and lower face is subtle.

Shell element edges

A (3D) shell element has a number of *edges* (or *sides*). The quadrilateral (Q4GS for example) has four edges, each with two nodes: for element I of Figure 1 these are IJ, JM, ML and LI.

At each edge of the shell, one or more other shells can be attached, which will be called *adjacents* (see Figure 2 and definition below). In flat structures only (at most) one adjacent can be present. In more geometrically complex structures several adjacents (the number is potentially unlimited) can be present along each edge at structural junctions. Adjacents have no equivalent in continuum elements, therefore a new data structure for the adaptive case in shells will have to be defined below.

The absence of any adjacents at a shell edge identifies that edge as a *free edge* of the shell structure. In other words, the edge lies on the “contour” of the shell structure.

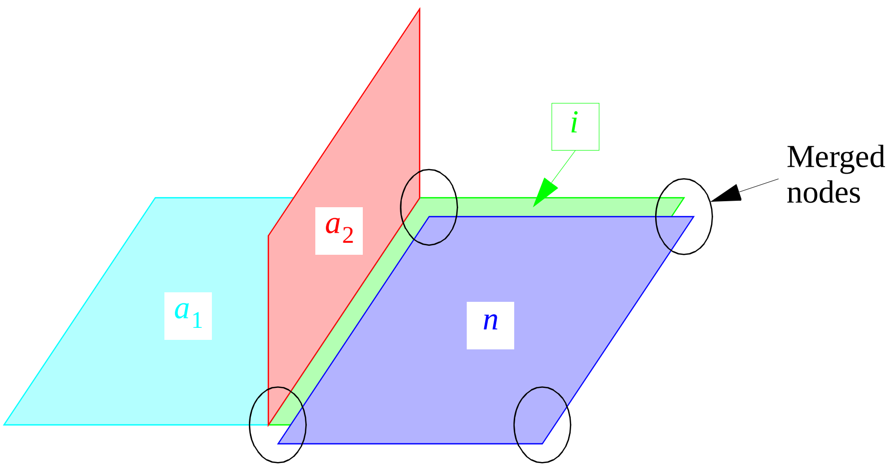


Figure 2 - Neighbors and adjacents of a Q4GS (non-adaptive case)

Example of neighbors and adjacents in the non-adaptive case

A simple example of neighbors and adjacents, restricting ourselves to the normal (non-adaptive) case for the moment, is shown in Figure 2. Let i be the shell element under consideration (in green). This element has the CLxx element n (in blue) as neighbor at one of its faces: note that the two elements are shown slightly displaced to illustrate the concept but in reality they are perfectly superposed and the nodes are merged (same node indexes).

Furthermore, element i has two adjacent elements along its left edge, the shell a_1 (in cyan) and the shell a_2 (in red), so that a “T” junction is present in the structure along this edge. Adjacents of a shell can only be other shells: continuum elements or elements of other more exotic classes (including CLxx) are not considered among the possible adjacents. The element and all its adjacents share the same (two) nodes defining the edge.

Returning now to the adaptive case, consider again Figure 1. When splitting element i, first each of its faces is considered, in analogy with the continuum case. If there is a neighbor across the face, which in the shell case can only be a CLxx element, then the neighbor is also split, exactly like in the case of neighboring CLxx to continuum elements.

Next, the edges of the shell are considered. The first edge is JK: along this edge there is no adjacent element (so the edge is on a free edge of the structural model) and therefore a new node b_1 has to be created. In analogy with the continuum case, we call such a node a *boundary-hanging* (or b-hanging, for brevity) node. A precise definition is given below. The knowledge of b-hanging nodes will be exploited, in analogy with the continuum case, to automatically propagate essential boundary conditions to newly created nodes along the boundary (more precisely, along the contour, or the free edges, of the shell structure). Together with each b-hanging node, the corresponding master nodes are identified, i.e. nodes J and K in the case of b_1. These are defined as the base nodes of the corresponding edge. The case of the next two edges KN and NM is similar, so two more b-hanging nodes b_2 and b_3 are created.

Finally, the fourth edge MJ has one adjacent element (j) with no children so, again, a new node h must be created. However, unlike the previous case, this node is set as *hanging* node, see details below. Like in the case of continuum, hanging nodes must have their degrees of freedom interpolated between those of the corresponding master nodes. In the case of node h, the master nodes are J and M (which both happen to be base nodes in this case, but this needs not be true in general).

With the definitions of faces and edges provided above, we are now ready to give more precise definitions of neighbors and adjacents for shells, valid both in the non-adaptive and in the adaptive case. In the adaptive case we will also have to consider pseudo-neighbours (like for continuum elements) and pseudo-adjacents (a new concept), which occur at locally non-conforming element-to-element interfaces. These will be abbreviated as p-neighbors and p-adjacents in the following.

First, the definitions of neighbor and p-neighbor for the continuum case (see reference [2] or [11]) are still valid for shells and can be stated as follows.

Definition of (regular) neighbor element in adaptivity

The neighbor of an element across a given face is the <u>same-level</u>, <u>active or inactive</u> element on the other side of the face, or 0 if there is no such element.

Definition of pseudo-neighbor element in adaptivity

The pseudo-neighbor of an (active) element across a given face is the <u>larger</u> (i.e. lower-level) <u>active</u> element on the other side of the face, or 0 if there is no such element. Inactive elements have no pseudo-neighbors.

As anticipated above, regarding neighbors to shells we will only consider specialized boundary-condition (CLxx) elements as possible candidates. Since CLxx elements are always split or unsplit automatically together with the neighboring element (be it continuum or shell), it follows that *p-neighbors to shell elements cannot exist*. The data structure for p-neighbors (`pseudo_ielfac`) will still be created for uniformity with other adaptive element types, but it will contain all zeroes in the case of shells.

The definitions of adjacents and p-adjacents are new and specific to shells.

Definition of (regular) adjacent elements to a shell in adaptivity

The adjacents of a shell element across a given edge are all the <u>same-level</u>, <u>active or inactive</u> shell elements sharing the same edge. A shell can have any number (including 0) of adjacents along each of its edges.

The adjacency relation (like the neighborship relation) is reciprocal: if element i has element j as one of the adjacents to one of its edges e, then j has i as one of the adjacents to one of its edges g.

Definition of pseudo-adjacent elements to a shell in adaptivity

The pseudo-adjacents of an (active) shell element along a given edge are all the <u>larger</u> (i.e. lower-level) <u>active</u> shell elements that (loosely speaking) "share" the same edge. More precisely, each p-adjacent p of a shell element i has the ancestor of i at the same level as p as adjacent along the edge considered. A shell can have any number (including 0) of pseudo-adjacents along each of its edges. Inactive shell elements have no pseudo-adjacents.

The p-adjacency relation (like the p-neighbor relation) is *not* reciprocal: if element i has element j as one of the p-adjacents to one of its edges e, then j can have i neither as one of the adjacents, nor as one of the p-adjacents, to *any* of its edges.

Note the following important differences between neighbors / p-neighbors and adjacents / p-adjacents, which of course have strong implications over the corresponding data structures:

- While (for continuum elements) the presence of a neighbor or of a p-neighbor across a face are mutually exclusive, a shell element can have *both* adjacents and p-adjacents along the same edge.
- While an element can have at most one neighbor or one p-neighbor for each face, a shell element can have both (any number of) adjacents and (any number of) p-adjacents along the same edge.

Example of adjacents and p-adjacents in adaptivity

For an example of adjacents and p-adjacents to a shell in adaptivity consider Figure 3 below. The upper part of the Figure shows the base mesh, which is similar to Figure 2 but without the CLxx element n for simplicity. In the bottom part of the Figure (third drawing), the element i has been split, giving four descendents i_1 to i_4. The element a_1 has not been split. The element a_2 has also been split, giving four descendents a_{21}, a_{22}, a_{23} and a_{24}.

As a consequence of splitting, the central nodes (indicated by a filled square) are always created without further checks, as anticipated above. Furthermore, six new b-hanging nodes (indicated by empty triangles) are created along an equal number of edges, which lie on the structure contour.

But here we want to focus on what happens along the common edge IJ. Let us assume that element i is split first (the order in which elements are split is irrelevant as concerns the final result) as shown in the central part of the Figure. We recognize that it has two adjacents and no p-adjacent along this edge. Therefore, a new node H is created which is hanging (solid circle) upon nodes I and J. If we then consider, for example, the descendent i_1, this will have no adjacents along its edge IH, and two p-adjacents (see definition above) a_1 and a_2 along the same edge. This is because edge IH is “part of” the bigger edge IJ so that, according to the definition of p-adjacents, both a_1 and a_2 have i (the ancestor of i_1 at the same level as a_1 and a_2) as (regular) adjacent across their edge IJ.

Then a_2 is split as shown in the bottom part of Figure 3. Along edge IJ, the a_2 element has no p-adjacents and two adjacent elements: a_1 (active) and i (which is currently inactive). Since the adjacent i has descendents at the same level as a_2 (the element being split), there is no need to create a new mid-edge node. This node (H) exists already and it must be identified and reused to set the connectivity of the descendents of a_2. Consider now what happens to the nature of node H, which was hanging before the splitting of a_2. In this case the node *remains hanging*, because there still exists a p-adjacent element (a_1) along the edge. After splitting a_2, the element i_1 has one adjacent (a_{21}) and one p-adjacent (a_1) along its edge IH.

The nature of node H would change from hanging to non-hanging (so that it is renamed N for clarity) only if, as shown in Figure 4, one would finally split also the element a_1. In this case therefore, no constraints would remain on this node and its dofs would become “free” (unconstrained) dofs of the structural model at all effects.

In this latter configuration, shown in Figure 4, the element i_1 has no p-adjacents and two (regular) adjacents (a_{21} and a_{12}) along its edge IN.

With all the above definitions and examples, it is now time to define the precise rules for the calculation of adjacents and p-adjacents in the adaptive case for shells. The rules for the construction of neighbors (there are no p-neighbors to shells, as already explained) are the same as for the continuum elements (see e.g. reference [11]) and are not repeated for brevity.

First of all it should be noted that in the base mesh, which is *conforming* by assumption, there may not exist any p-neighbors, nor any p-adjacents. Therefore, at the beginning of the calculation it is sufficient to compute the neighbors (exactly like in the case of continuum elements) and the adjacents. Then, the neighbors and the adjacents will be continuously updated during the mesh adaptation process, while the p-adjacents (no p-neighbors to shells can exist as already explained) will be computed during the mesh adaptation process.

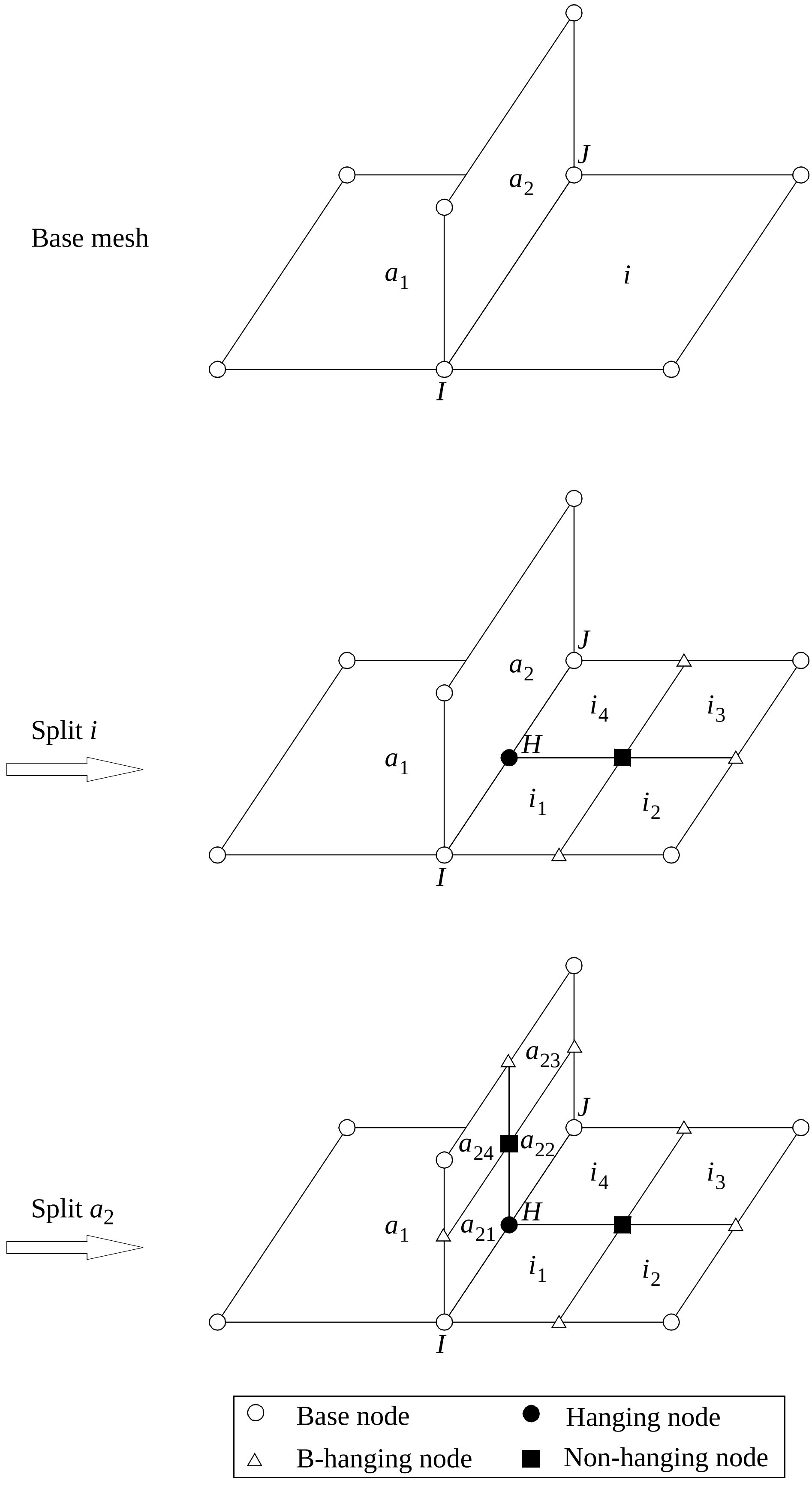


Figure 3 - Adjacents and p-adjacents of a Q4GS in the adaptive case

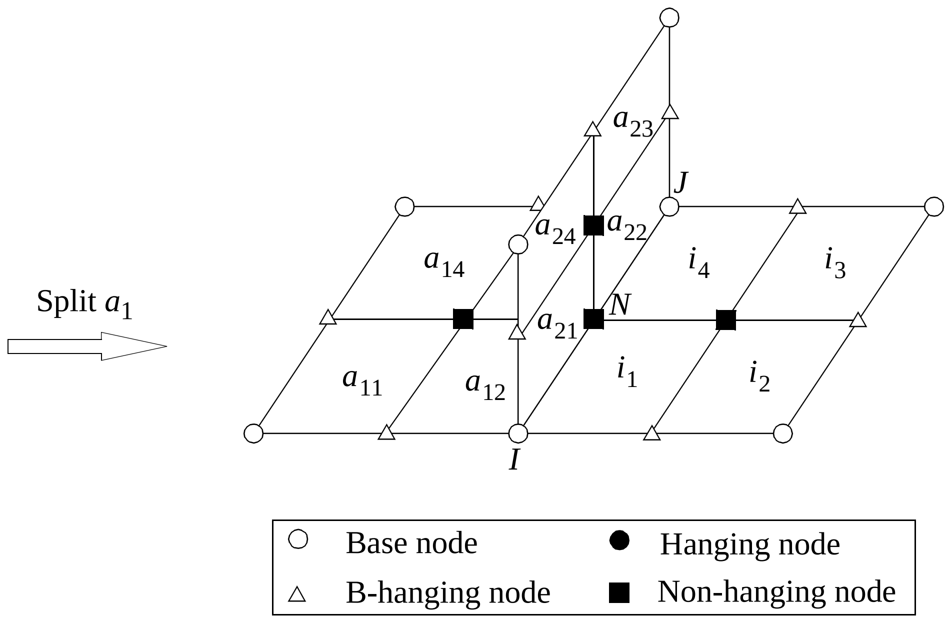


Figure 4 - Adjacents and p-adjacents of a Q4GS in the adaptive case (cont'd)

2.1 Data structure for the adjacents

From the data structure viewpoint, the main difference between the neighbors / p-neighbors and the adjacents / p-adjacents is that the number of the first is fixed and known a priori (exactly one, possibly empty, for each face of an element), while the number of the second is arbitrary (zero or more).

To accommodate such a data structure, we build the following quantities (in a new module `M_ADJACENTS`):

- A new array `INTEGER, POINTER :: P_ELEDGE(IEL)` is a table containing a pointer (address) to the first edge of element `IEL` in the following table `ELADJAC`, to be described next. The length (dimension) of the `P_ELEDGE` table is fixed and equal to `NELEM+1`, where `NELEM` is the total number of elements. This allows to compute the number of edges of element `IEL` by difference: `NEDGE = P_ELEDGE(IEL+1) - P_ELEDGE(IEL)`. Alternatively (and perhaps more cleanly) one can dimension `P_ELEDGE` to `NELEM` and define the number of edges of each element type in a new "element characteristics" table, say `NCEL2()` in the include `CAREL.INC`, initialized in `INICO1.FF`.
- A new derived data type `TYPE ADJAC`, containing a list (`ADJACENTS`) of all the adjacent elements to one edge of an element. We assume to keep this list *in growing order* to facilitate searches and comparisons:

```
TYPE ADJAC
```

```
    INTEGER :: N_ADJACENTS
    INTEGER, POINTER :: ADJACENTS(:)
  END TYPE ADJAC
```

- A new array `TYPE(ADJAC), POINTER :: ELADJAC(:)` containing all the lists of adjacents. Therefore, to access the number `NADJ` and the list `ADJ(:)` of adjacents, say, to *the second* edge of element `IEL`:

```
  IAD = P_ELEDGE(IEL) + 1 ! plus one because we want edge #2
  NADJ => ELADJAC(IAD)%N_ADJACENTS
  IF (NADJ > 0) ADJ => ELADJAC(IAD)%ADJACENTS
```

Since the number of adjacents to each edge of an element and the indexes of such adjacents vary during the mesh adaptation process, the use of a pointer list provides all the needed flexibility. The definition of `N_ADJACENTS` is also introduced in the derived type to avoid using the (potentially expensive) `SIZE()` function to determine the length of the list, and the preliminary use of `ASSOCIATED()` to see whether the list is allocated or not (`SIZE` cannot be used on a non-allocated list).

The calculation of adjacents *for the base mesh* is done in the initialization phase by a loop over the elements, in which only shell elements are considered. For each edge of an element (which has two nodes, in 3D), all other shell elements sharing the same edge (i.e. the same couple of nodes) are sought. The order in which the nodes appear in the edge is irrelevant.

During the transient, the table of adjacents is constantly updated (locally) as part of the element splitting and unsplitting process. The complete table is never recomputed for obvious efficiency reasons.

2.2 Data structure for the p-adjacents

To accommodate the p-adjacents, the same table `P_ELEDGE(:)` and the same derived data type `TYPE ADJAC` defined above are used, since they are suitable both for adjacents and for p-adjacents.

In addition, we build the following quantity (in the same new module `M_ADJACENTS` used for the adjacents):

- A new array `TYPE(ADJAC), POINTER :: ELPADJAC(:)` containing all the lists of p-adjacents. Therefore, to access the number `NPADJ` and the list `PADJ(:)` of p-adjacents, say, to *the second* edge of element `IEL`:

```
  IAD = P_ELEDGE(IEL) + 1 ! plus one because we want edge #2
  NPADJ => ELPADJAC(IAD)%N_ADJACENTS
  IF (NPADJ > 0) PADJ => ELPADJAC(IAD)%ADJACENTS
```

In the base mesh there are no p-adjacents, so only the part of the data structure common to both adjacents and p-adjacents is built in the initialization phase.

During the transient, the table of p-adjacents is constantly updated (locally) as part of the element splitting and unsplitting process.

The precise rules for computing or updating the adjacents and the p-adjacents during an element splitting or unsplitting are detailed in Sections 2.3.2 and 2.4.2 below, respectively.

2.3 Element splitting algorithm for the Q4GS

We present now the refinement algorithm, which is similar to the one detailed in Section 2.7.2 of reference [1] for the 2D 4-node quadrilateral, but we consider here the adjacents and the p-adjacents of the shell element along each of its edges, rather than the neighbor or p-neighbor (mutually exclusive) of the continuum element along each of its "faces" (this was the terminology indicating the edges or sides for 2D continuum elements). The element i being refined is generic in the following algorithm, i.e. it can be either a base element of the mesh, or a descendent element (at any level).

2.3.1 Updating of the nodes nature

When splitting an element, some new nodes are created. Their nature (non-hanging, hanging or b-hanging) must be determined. In addition, some nodes are reused and their nature must be re-determined since it might change because of the splitting. Here is the detailed algorithm:

- Upon refinement of element i, the new central node is always created as "non-hanging".
- Then, the four element edges of i are inspected in turn. For each edge, we read all the adjacents (if any) and all the p-adjacents (if any) from the adjacency tables.
- If there are no adjacents and no p-adjacents, then the edge is on the contour of the shell structure and a new mid-edge node must be created. The new node is set as b-hanging upon the two nodes of the *base edge* to which it belongs, so that it will inherit any essential boundary conditions prescribed on such nodes. The base edge is the edge of the base mesh that "contains" the edge under consideration. Since edge numbering is preserved upon element splitting and un-splitting, the base edge is simply the edge of the *base ancestor* of i (the ancestor of i at zero, or base, level) with the same index as the current edge.
- Else if there are only p-adjacents, a new mid-edge node must be created. The new node is set as hanging upon the two nodes of the current edge.
- Else there are some adjacents (and possibly also some p-adjacents). We inspect the adjacents. If all adjacents are leaves (i.e. they have no children) then a new mid-edge node must be created as

hanging upon the two nodes of the current edge. Else at least one adjacent is a branch (it has children), so that the mid-edge node exists already. This node must be identified and should be reused in the splitting process. As concerns the nature of the node: it must have been hanging prior to splitting of the current element (and this should be checked for consistency). Then, if there are only adjacents and all adjacents have children, then the node becomes non-hanging, otherwise (i.e. if there is at least one adjacent without children, or at least one p-adjacent) the node remains hanging.

2.3.2 Adjacents and p-adjacents update during element splitting

When splitting element i, of course we must compute the adjacents and the p-adjacents of its children (which did not exist prior to splitting).

Neither the adjacents nor the p-adjacents of i itself are affected by the splitting of i. However, the splitting of element i has potential effects on the adjacents and p-adjacents of its "surrounding" elements, namely of all elements which had i either as adjacent or as p-adjacent.

For the first type of surrounding elements, those having i as adjacent, nothing needs to be done: in fact, although it is being split, i remains adjacent (although it becomes inactive).

The second type of surrounding elements instead, those having i as p-adjacent, must be inspected more in detail and are appropriately treated in the algorithm detailed in the next paragraph. Such elements are to be sought among all descendents (at any level, active or not) of i 's adjacents.

2.3.3 Adjacents and p-adjacents of the children and of the surrounding elements

Assume we are splitting an element i at level L (so that its children will be at level $L + 1$). We consider each child i_k in turn and loop over the edges e of the child (for the Q4GS it will be $k = 1, \dots, 4$ and $e = 1, \dots, 4$).

- If the edge e is "internal" to the patch formed by the children, then there is just one adjacent element (one of the other three children), which is easily identified, and no p-adjacents. In fact, there can be no junctions in the interior of an element.
- Else, the edge e is "external" to the patch formed by the children and belongs to an edge of the (parent) element i which is being split. In this case, we search all adjacents a_k (if any) and all p-adjacents p_m (if any) of the parent element i along the concerned edge, by noting that the same orientation is kept in the children as in the parent, so that the local edge number is the same for both.

I *If there are neither adjacents nor p-adjacents the parent's edge is a free edge and such it will remain for the child, which will have no adjacents and no p-adjacents along this edge.*

II *Else if there are only p-adjacents then the child will have the same p-adjacents and no adjacent along the edge under consideration.*

III *Else there are one or more adjacents and zero or more p-adjacents. All p-adjacents (if any) of the parent become also p-adjacents of the child under consideration along this edge. Then we loop over all the adjacents of i along the current edge.*

* *If an adjacent is active (and thus has no children of its own), then it becomes an (additional) p-adjacent of the child along this edge.*
* *Else it is inactive, thus it has its own children (at the same level $L+1$ as the child of i under consideration). Exactly one of these children (which must be determined by nodes comparison), say a_j, becomes an adjacent of the child i_k under consideration (and reciprocally, since the adjacency relationship is symmetric). Let g be the edge of a_j along which it has i_k as adjacent. Since i is being split, we remove it from the list of p-adjacents of a_j along its edge g, after checking (for consistency) that it was there. Finally, we search all descendents of a_j, at any level Λ (where it is of course $\Lambda > L+1$), both active and inactive, which are "attached" to edge g of a_j, and select those which had i as p-adjacent along their edge g: we remove i from the list of p-adjacents and add i_k to the list of p-adjacents of such elements along their edge g.*

2.4 Element un-splitting algorithm for the Q4GS

We present now the un-refinement algorithm, which is similar to the one detailed in Section 2.7.4 of reference [1] for the 2D 4-node quadrilateral, but (like in the case of element splitting) we consider here the adjacents and the p-adjacents of the shell element along each of its edges, rather than the neighbor or p-neighbor (mutually exclusive) of the continuum element along each of its "faces" (this was the terminology indicating the edges or sides for 2D continuum elements). The element i being un-refined is generic in the following algorithm, i.e. it can be either a base element or a descendent element (at any level).

2.4.1 Updating of the nodes nature

When un-splitting an element, some old nodes are destroyed. In addition, the nature (non-hanging, hanging or b-hanging) of all nodes involved in the unsplitting process and which do remain in existence must be re-determined, since it might change because of the un-splitting. Here is the detailed algorithm:

- Upon un-refinement of element i (i.e. upon suppression of its four descendents i_k, $k = 1, \ldots, 4$), the old central node is always non-hanging (check) and must be simply suppressed. Its slot in the nodes table becomes free for reuse in successive mesh refinements.
- Then, the four element edges e of i are inspected in turn. For each edge, we read all the adjacents (if any) and all the p-adjacents (if any) from the adjacency tables.

- If there are neither adjacents nor p-adjacents, then the parent's (i) edge is a free edge (on the contour of the shell structure). Therefore, the old mid-edge node must be suppressed, after checking (for consistency) that this node was b-hanging.
- Else if there are only p-adjacents, then the old mid-edge node must be suppressed, after checking (for consistency) that this node was hanging.
- Else there are some adjacents (and possibly also some p-adjacents). We inspect the adjacents. If all adjacents are leaves (i.e. they have no children), then the old mid-edge node must be suppressed after checking (for consistency) that it was hanging (upon the two nodes of the current edge). Else at least one adjacent is a branch (it has children), so that the old mid-edge node must remain and becomes (or remains) hanging upon the two nodes of the current edge, after checking (for consistency) its nature before the unsplitting of i. If either there is at least one p-adjacent, or there is no p-adjacent but at least one of the adjacents is a leaf (has no children), then the mid-edge node must have been (already) hanging prior to un-splitting of the current element. Otherwise (i.e. if there is no p-adjacent and all adjacents are branches, i.e. have children), the mid-edge node must have been non-hanging.

2.4.2 Adjacents and p-adjacents update during element unsplitting

Like in the case of element splitting, neither the adjacents nor the p-adjacents of i are affected by the unsplitting of i, but we must update the adjacents and p-adjacents of the elements surrounding i 's children (i.e. any elements having one of i 's children either as adjacent or as p-adjacent).

In addition, of course, all adjacents and p-adjacents of the children i_k of i are destroyed (together with the children themselves) at the end of the unsplitting process.

2.4.3 Adjacents and p-adjacents of the children and of the surrounding elements

Assume we are unsplitting an element i at level L, so that its children i_k are all active and at level $L + 1$.

We loop over each edge e of i (for the Q4GS $e = 1, \ldots, 4$) and identify the indexes of the two children, say i_a and i_b, which are "attached" to edge e (this is immediate).

- Then, to find out all elements having either i_a or i_b either as adjacent or as p-adjacent, we loop over all the adjacents of i along the edge e.

  I *If the adjacent is active, i.e. it has no children of its own, we skip it.*

  II *Else the adjacent is inactive. We search all its descendents, both active and inactive, and at any level Λ (where of course it must be $\Lambda > L$), and select those which have i_k (with $k = a$ or $k = b$) as either adjacent or p-adjacent along one of their edges g.*

* *If such a descendent has level $\Lambda = L + 1$, then it must have had i_k as (regular) adjacent. Then, it will no longer have i_k as adjacent, but it will instead have i as p-adjacent.*
* *Else, the descendent has level $\Lambda > L + 1$: then it must have had i_k as p-adjacent. In this case it will no longer have i_k as p-adjacent, but it will instead have i as p-adjacent.*

The children of i are destroyed at the end of unsplitting operations and their adjacents and p-adjacents lists are simply erased. The i element becomes active again.

2.5 Nodal degrees of freedom

With the implementation of shells in adaptivity, the problem of an appropriate treatment of nodal degrees of freedom (dofs) in the adaptive case arises. Shells have more dofs than continuum elements because they have rotations in addition to translations.

2.5.1 Non-adaptive case

The *nodal coordinates*, `XGLOB` for the current ones and `XINIT` for the initial ones, are stored in module `M_NODAL_VARS`. They have the *same length for each node*, equal to `IDIM`, the space dimension, which is usually 2 or 3. Therefore they can typically be declared as:

```
REAL(8) :: XGLOB(IDIM,*)
```

The first index of the table indicates the spatial component, the second one indicates the node.

Nodal variables are contained in arrays (`DFX` for displacement, `V` for particles velocity, etc.) that are also stored in module `M_NODAL_VARS`. However, each node in the mesh can have *a different number of dofs*. This is because different types of elements in the mesh can require different numbers of dofs at their nodes, and such elements are connected at (common) nodes. Thus for example, a node common to a continuum element, which requires 2 dofs in 2D, and to a shell element, which requires 3 dofs (2 translations plus one rotation) in 2D, will require 3 dofs (the maximum among those of the concerned elements).

In order to access the dofs of a generic node, use is made of a pointer (integer table) `POSP`, also stored in module `M_NODAL_VARS`. This array is declared as:

```
INTEGER :: POSP(*)
```

and is built in such a way that `POSP(I)` is the address of the first dof of node `I` in the tables `DFX`, `V` etc. of the nodal quantities. The *number of dofs* `NDOFI` *of the generic node* `I` can therefore be obtained by difference:

```
NDOFI = POSP(I+1) - POSP(I) ! n. of dofs of node I
```

In order for the above to work also for the last node (`NPTL`) in the mesh, the `POSP` table is allocated for `NPTL1 = NPTL + 1` nodes, where `NPTL` is the total number of nodes in the mesh. The index of the last dof of the last node in the mesh (i.e. the *total number of dofs*) is thus `POSP(NPTL1) - 1`.

In the non-adaptive case, the `POSP` table is built in the `GEOM` subroutine by associating to each node the maximum number of dofs among those required by the elements attached to that node. As an example, consider the following cases, illustrated in Figure 5.

In the left part of the Figure, we have a mesh (`POSP01`) composed of two elements: a cube with 8 nodes and a shell with 4 nodes. the two elements are not connected so a total of 12 nodes is required. The cube nodes require 3 dofs (translations) each, while the shell nodes require 6 dofs (translation plus rotations) each. With the numbering of nodes shown in the Figure, the `POSP` table is as follows:

```
POSP = 1,4,7,10,13,16,19,22,25,31,37,43,49
```

We see therefore that (as expected):

- The nodes 1 to 8 (those belonging to the cube) have 3 dofs each, for a total of 24 dofs (1-24).
- The nodes 9 to 12 (those belonging to the shell) have 6 dofs each, for a total of 24 dofs (25-48).
- The `POSP` table contains 13 entries (for a total of 12 nodes). The total number of dofs is given by the last entry in `POSP` minus 1, thus $49 - 1 = 48$.

In the right part of Figure 5 we have a mesh (POSP02) containing the same elements as before, but now connected so that only 10 nodes are required. In fact, nodes 2 and 5 are common to both the cube and the shell.

With the numbering of nodes shown in the Figure, the `POSP` table is as follows:

```
POSP = 1,4,10,16,19,25,31,34,37,40,43
```

We see therefore that (as expected):

- The nodes belonging only to the cube (nodes 1, 4, 7, 8, 9 and 10) have 3 dofs each.
- The nodes belonging only to the shell (nodes 3 and 6) have 6 dofs each.
- The nodes belonging to both the cube and the shell (nodes 2 and 5) have 6 dofs each.
- The `POSP` table contains 11 entries (for a total of 10 nodes). The total number of dofs is given by the last entry in `POSP` minus 1, thus $43 - 1 = 42$.

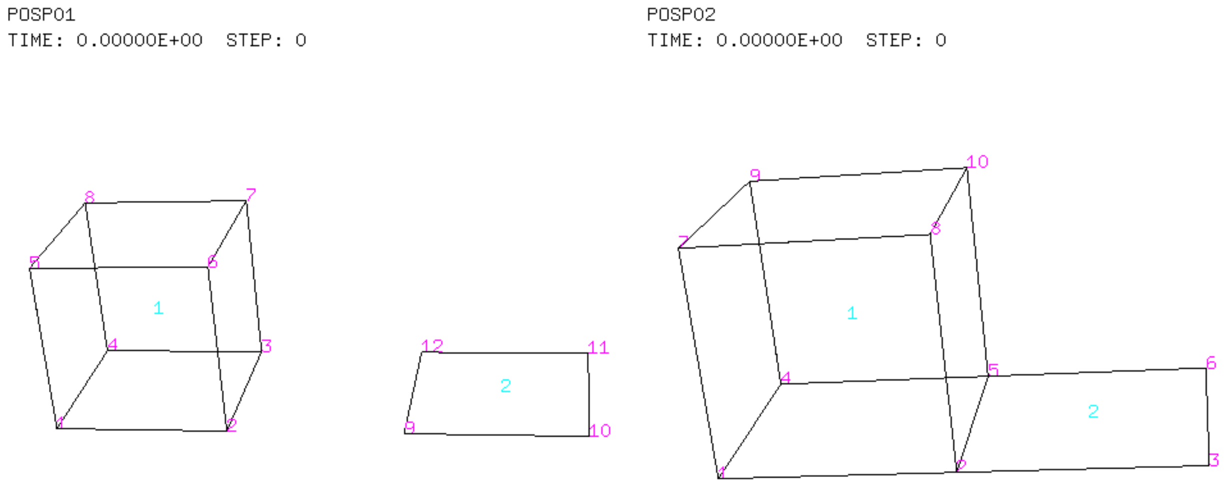


Figure 5 - Meshes for checking the construction of POSP (non-adaptive case)

2.5.2 Adaptive case

In case of adaptivity, additional nodes to be created during the mesh refinement process are allocated in the so-called extension zone of the memory, see reference [1]. These are called *descendent nodes*, to distinguish them from the *base nodes*, those belonging to the base (unrefined) mesh provided by the user.

While all base nodes are always used, at any level of refinement of the mesh, a descendent node can or cannot be used, depending on the particular instant of the transient considered. In addition, since elements are continuously refined and un-refined, *a descendent node can belong to different elements* during the transient.

Therefore, in principle a "static" (constant) dof pointer table `POSP` cannot be built in the adaptive case. Such a table would need to be continuously updated, but this would require the corresponding updating of all vectors of nodal quantities that are pointed to by `POSP` (i.e. `DFX`, `V`, etc.). But this process would be too expensive in terms of CPU.

Therefore, in adaptive cases we resort to a simplification, which however requires some more memory (for the nodal dofs tables, such as `DFX`, `V` etc., not for `POSP` itself). A static (constant) `POSP` is built also in adaptive cases, according to the following strategy:

- The (initial) part of `POSP`, related to base nodes, is built by using the exact number of dofs required for each node, exactly like in the non-adaptive case.

- The (second and last) part of `POSP`, related to descendent nodes, is built by using a *constant number of dofs for each descendent node*, equal to the *maximum* number of dofs required by any of the adaptive element types present in the calculation. These element types are known at the moment of calling `GEOM` (which builds up `POSP`), since they must be declared in the dimensioning part of the input file (`DIME ADAP` directive). In this way, any descendent node can become attached to (belong to) any descendent (adapted) element during the course of the transient calculation.

Of course, this means that in general there will be some redundant dofs in the extension part of the tables. All variables for such dofs should be made equal to zero as part of the mesh refinement and unrefinement process.

2.6 The T3GS shell element

The T3GS shell element is the triangular 3-node equivalent of the quadrilateral (Q4GS) 4-node shell and uses the same formulation. Therefore, all the algorithms presented so far for element splitting and unsplitting can be applied also to the T3GS, by just making suitable adjustments in the number of nodes (3 instead of 4), the number of edges (3 instead of 4) and the number of Gauss Points (just one point per lamina instead of four points per lamina).

2.7 The 2-node segment-like elements

Finally, we consider 2-node (segment-shaped) elements. In 2D, such elements can represent shells. They can also represent beams or bars, either in 2D or in 3D.

A partial (preliminary) implementation of adaptivity and of error indicators for the FUN2 and FUN3 elements had been already presented in reference [5]. This had been tested in the 2D case (FUN2) on a 1-D wave propagation problem (modified Verdugo's test), see Section 6 of reference [5] and test case VE1D16 therein. The solution showed some oscillations and did not respect the symmetry of the problem. The model did not account for the possible presence of "branchings" in the shell, beam or bar structure. Only one "neighbor" (or "p-neighbor") was allowed at each edge of the 2-node element.

Here the implementation for FUN2 has been generalized by recognizing the similitude between these segment-shaped (SEG2) elements and the 3D shells considered in the previous paragraphs. Like in the case of shells, these elements can have "true" neighbors (but no p-neighbors), but these neighbors are also 2-node segment-like elements and can only be of the CLxx type, used to prescribe some boundary conditions, e.g. an external pressure acting on the shell/beam/bar.

In addition, SEG2 elements do have in general a number of adjacents (and of p-adjacents, in adaptivity), exactly like 3D shells, at each one of the element edges. However, an edge contains just one node and not two nodes. The situation in the non-adaptive case and the nomenclature is sketched in Figure 6.

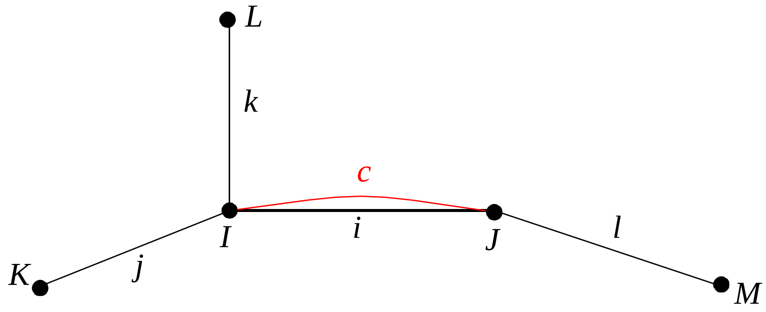


Figure 6 - Neighbors and adjacents of a SEG2 shaped element (non-adaptive case)

Let i be the SEG2 element (e.g., a 2D shell) under consideration. This element has nodes I and J. The element has two *faces*: the segment IJ and the segment JI. It also has two *edges*: the point I and the point J.

Neighbor elements, if present, are attached to the element faces. For example, we have considered a CLxx element c, which also has the nodes I and J (shown slightly deformed for clarity).

Adjacent elements, if present, are attached to the edges. In the example, there are two adjacents (j and k) along the first edge of i and only one adjacent (l) along the second edge. If there are no adjacents, the edge is considered to be "on the contour" of the structure.

In the adaptive case, as elements are split and unsplit, a SEG2 element can have any number (including zero) of adjacents and any number (including zero) of p-adjacents along each one of its edges, in full analogy with the 3D shell elements considered previously. The treatment of these is similar to the case of 3D shells and is not further detailed here.

3. Numerical examples

We present some numerical examples in order to test the algorithms described in the previous Sections.

3.1 Checking element splitting / unsplitting

The first examples are just simple checks of element splitting. We start with a very coarse base mesh, then progressively refine it by the `ADAP SPLI` interactive command and finally unrefine it by the `ADAP USPL` command. The purpose is to check the creation and the evolution of hanging and b-hanging nodes as splitting proceeds. The creation and suppression of adjacents and b-adjacents is also checked.

The calculations are summarized in Table 1.

Case	Base Mesh	Notes	Steps	CPU [s]	Els*step
Q4GS01	2 Q4GS	Progressive refinement / unrefinement by ADAP SPLI / USPL	4	—	—
Q4GS02	3 Q4GS	Progressive refinement / unrefinement by ADAP SPLI / USPL	6	—	—
Q4GS03	100 Q4GS	2 WAVE directives MAXL 4 to adapt the mesh	166	4.1	268,058

Table 1 - Calculations to check element splitting / unsplitting

Q4GS01

In this test we start with two quadrilateral shells attached at one edge (see top part of Figure 7), then we split one of then, and finally split one of its descendents (see central and bottom parts of Figure 7). Of course, the base mesh is conforming and there are no hanging and no b-hanging nodes.

Upon splitting of element 1 (central part of the Figure), five new nodes (7 to 11) are created, and four new elements (3 to 6) are created. One of the new nodes (node 8) is hanging upon nodes 2 and 5, because it lies on an internal non-conforming edge. Three of the new nodes (7, 9 and 10) are b-hanging (upon *base* nodes 1-2, 5-4 and 4-1, respectively) because they lie on edges which are along the contour of the structure.

Then, upon splitting of element 4 (one of the descendents of element 1 created during the first splitting operation), again five new nodes (12 to 16) are created, and four new elements (7 to 10) are created. Three of the new nodes (13, 14 and 15) are hanging (upon nodes 2-8, 8-11 and 11-7, respectively) because they lie on non-conforming internal edges of the structure. One new node (12) is b-hanging (upon *base* nodes 1-2) because it lies on an edge which is along the contour of the structure.

Concerning hanging and b-hanging nodes, note that:

- Hanging nodes may hang (depend) upon other hanging nodes (at *any level* in the adaptive hierarchy). For example, in the last part of Figure 7 node 13 hangs upon nodes 2 and 8. While node 2 is a base (and therefore non-hanging) node, node 8 is a level-2 node which in turn hangs upon nodes 2 and 5 (which both happen to be base nodes in this particular case). This is not a problem, since the hanging relationships are solved implicitly in a system of constraints, together with all other essential boundary conditions of the problem, and therefore all conditions are satisfied simultaneously. Hanging nodes can change their status to non-hanging nodes (and possibly also back to hanging) during the course of their life.
- B-hanging nodes always hang (depend) upon *base* nodes, which are called the master nodes of the b-hanging node. Each b-hanging node automatically inherits any essential boundary conditions which are common to *all* its master nodes. Therefore, some constraints *may* be applied to b-hanging nodes, in addition to hanging nodes. B-hanging nodes always remain b-hanging (and upon the same set of master base nodes) during the entire course of their life.

By unsplitting (in this order) first element 4 and then element 1, the mesh returns in its original (base) configuration, and this is checked in the example.

Q4GS02

In this test we start with three quadrilateral shells attached along one common edge (see top part of Figure 8), i.e. with the simplest possible case of shell structure with a junction. Then we split each one of them in turn and check the appearance of hanging and b-hanging nodes, see Figures 8 and 9. Note that node 8 becomes hanging after the splitting of element 2, it remains hanging after the splitting of element 3 (because of the presence of a still unsplit element, element 1, in its adjacency), and then returns non-hanging after element 1 has been split.

By unsplitting the elements in the reverse order in which they were split, the mesh returns in its original (base) configuration, and this is checked in the example.

Q4GS03

This test is similar to case TEST10 of reference [1] but uses the 3D shell element Q4GS instead of the 2D continuum element Q41L. A square plate is meshed coarsely by 10 elements along each direction, resulting in a base mesh of 100 elements. Then, two `WAVE` directives of type `SPHE` are used to progressively refine the mesh up to `MAXL 4` and then unrefine it.

The mesh checks only geometric aspects of the adaptive process since no load (and thus no stress) is applied to the plate. The progressive mesh adaptation is shown in Figure 10.

3.2 Elastic oscillations of a plate

The next examples are physical and concern the elastic oscillations of a square plate. The plate is similar to the one already used in test Q4GS03. It has a length of 10 m, a thickness of 0.1 m and is completely clamped (all dofs blocked) along its contour. Two versions of the problem are studied. In the first one, the plate has an initial velocity of 100 m/s in the normal direction to the plate itself z axis), while in the second one the plate is loaded by a uniform pressure of 10 bar ($1.\times10^6$ Pa), constant in time. The material is linear elastic.

The calculations are summarized in Table 1.

Case	Base Mesh	Notes	Steps	CPU [s]	Els*step
Q4GS04	400 Q4GS	Non-adaptive solution with INIT VITE	629	1.2	252,000
Q4GS05	100 Q4GS	Adaptive solution with INIT VITE (INIT ADAP SPLI LEVE 2)	629	1.5	252,000
Q4GS06	100 Q4GS	2 WAVE directives MAXL 2 to adapt the mesh	334	0.5	36,746
Q4GS07	1 Q4GS 1 CL3D	Shell and CLXX are numbered in opposite orders	4	—	—
Q4GS08	1 Q4GS 1 CL3D	Shell and CLXX are numbered in the same order	4	—	—
Q4GS09	400 Q4GS	Non-adaptive solution with IMPE PIMP	751	1.9	601,600
Q4GS10	100 Q4GS	Adaptive solution with IMPE PIMP (INIT ADAP SPLI LEVE 2)	751	2.2	601,600

Table 2 - Calculations of the elastic oscillations of a plate

Q4GS04

This is a non-adaptive solution of the plate problem, with a uniform mesh of $20 \times 20 = 400$ Q4GS elements. The initial velocity is applied to all nodes of the plate except those on the contour, which are clamped. Figure 11 shows the displacement of the central node of the plate and is assumed as a reference for the following (adaptive) solution.

Q4GS05

This test is similar to Q4GS04 but uses a coarser (base) mesh of only 100 elements. At step 0, the mesh is refined (once) by the INIT ADAP SPLI LEVE 2 directive, so that 400 elements are obtained, like in case Q4GS04. The initial velocity is applied to all (base) nodes except those on the contour.

Upon mesh adaptation, the velocity of the descendent nodes is obtained by linear interpolation of the velocities of the surrounding base nodes. This results in an initial velocity distribution which is slightly different from that of case Q4GS04 in the first row of (adaptive) nodes near the contour. For this reason, we cannot expect that the results of this calculation be identical to those of the reference (non adaptive) case. But they should be quite similar.

Figure 12 compares the displacements in the two tests Q4GS04 and Q4GS05, showing small but visible differences (as expected). Tho check that such differences come from the different initial velocity distribution, we repeat the same tests by specifying the initial velocity on all nodes (including the clamped ones). This results in very poor energy checks, since a lot of energy is dissipated in abruptly blocking the clamped dofs with non-zero initial velocities at the first half-step (recall that velocities are imposed at mid-step in the code). However, in this case the two results become identical, as shown in Figure 13.

Q4GS06

This test is similar to Q4GS05 but uses a different adaptation strategy. Instead of refining the whole mesh at step 0 and then keeping it constant until the ene, we start from a coarse mesh of 100 elements and then progressively refine and un-refine it by two `WAVE` directives like in case Q4GS03. Of course, these waves have nothing to do with the physics of this problem (oscillations of the plate). The purpose is to show that the solution of the oscillation problem is not (too much) disturbed by a practically random (since it has nothing to do with the physics of this problem) mesh refinement and unrefinement.

The displacements of the central node are shown in Figure 14. The vertical displacement is different from the one of case Q4GS04 (as expected) but it has the same overall shape. Note that the solution remains symmetric since the two in-plane displacements are both zero.

Figure 15 compares the solutions of cases Q4GS04 and Q4GS06.

Q4GS07 and Q4GS08

These two are simple tests to check the automatic splitting and un-splitting of a CLXX element (Cl3D in this case) attached to a Q4GS adaptive element. As the shell element is refined or unrefined, the CLXX element must follow the same pattern. When a CLXX element is attached to a continuum element, its orientation with respect to the element is guaranteed to be such that the normal to the CLXX points towards the continuum. If this requirement is not satisfied in the mesh as supplied by the user, the code inverts the numbering of the CLXX element so as to comply with this rule.

However, the same rule cannot be enforced when a CLXX is attached to a shell element, because the shell element has two faces, whose normals point in opposite directions. In this case, any orientation of the CLXX element is accepted by the code and not modified.

When splitting or unsplitting a CLXX element attached to a shell, only the neighbors and p-neighbors (like in the case of continuum elements) play a role. The adjacents and p-adjacents along the edges of the shell are not playing any role.

Note that such adjacents and p-adjacents are (other) shell elements. A CLXX element attached to a shell is not considered an adjacent of the shell.

In the first test (Q4GS07) the CLXX element is oriented in the opposite sense with respect to the numbering of the shell element. In the second test the CLCCXX element is oriented in the same sense. Results are consistent. The CLXX elements are tested more thoroughly in the next two cases, where they are used to apply a pressure to the plate.

Q4GS09

This test is similar to Q4GS04 (non-adaptive solution) but uses an applied pressure rather than an initial velocity to load the plate. Thus, the ambiguities related to the initial distribution of the velocity observed in cases Q4GS04 and Q4GS05 are avoided.

The solution in terms of central displacement of the plate are shown in Figure 16.

Q4GS10

This test is similar to Q4GS09 but starts from a coarse 100-element base mesh which is adapted to level 2 at step 0. The solution is compared in Figure 17 with the reference (Q4GS09) showing perfect agreement.

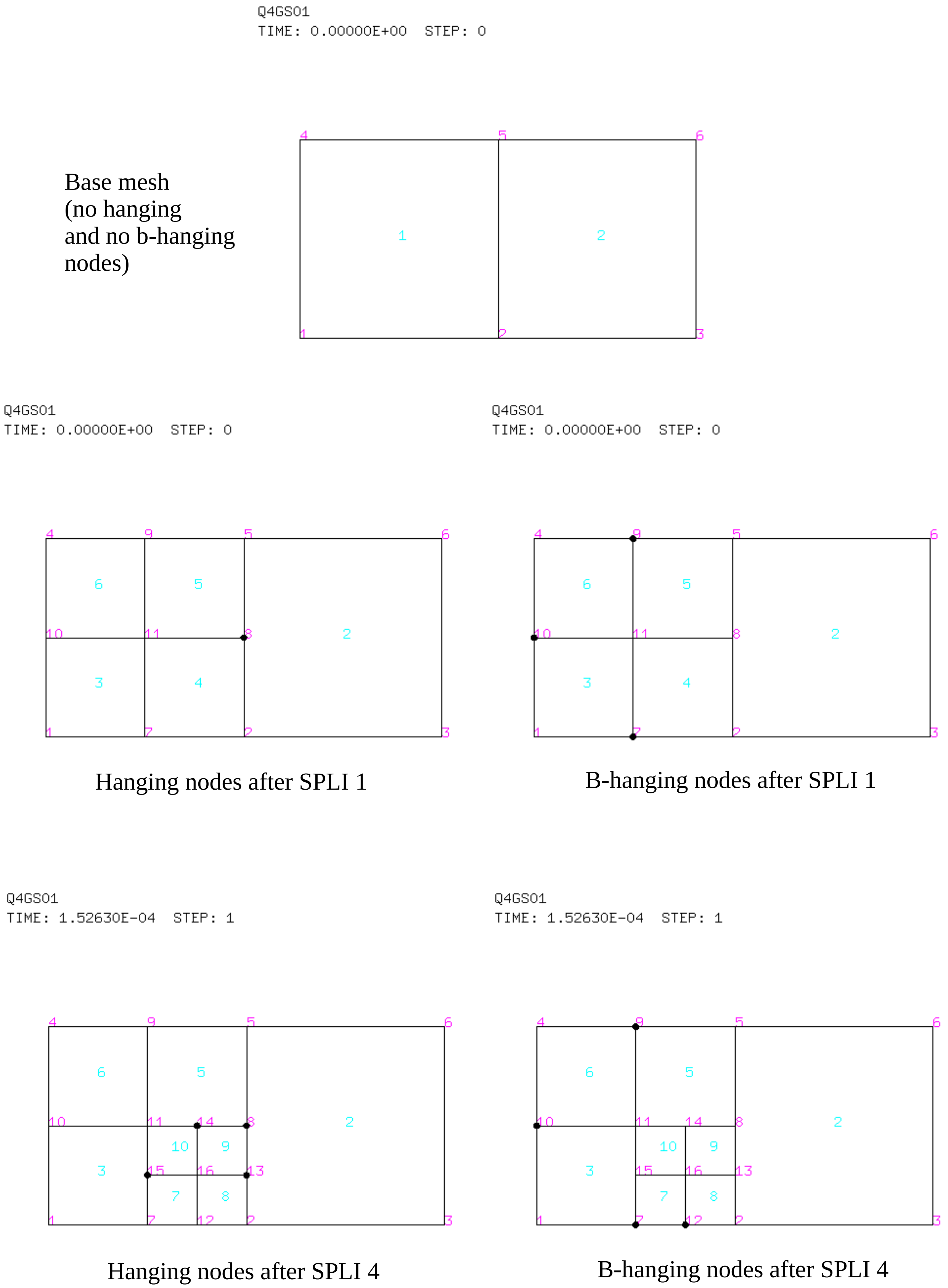


Figure 7 - Progressive mesh refinement in case Q4GS01

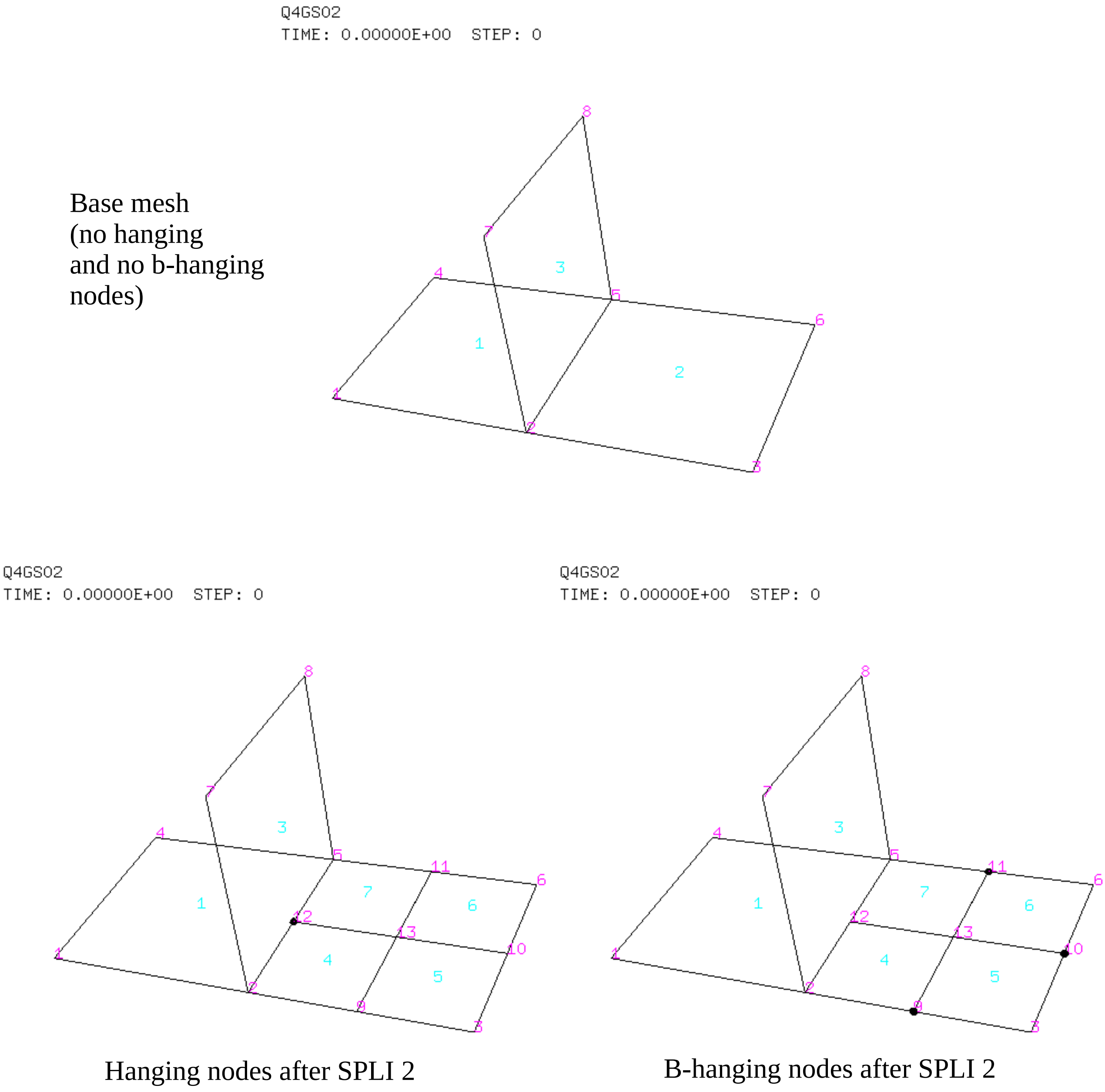
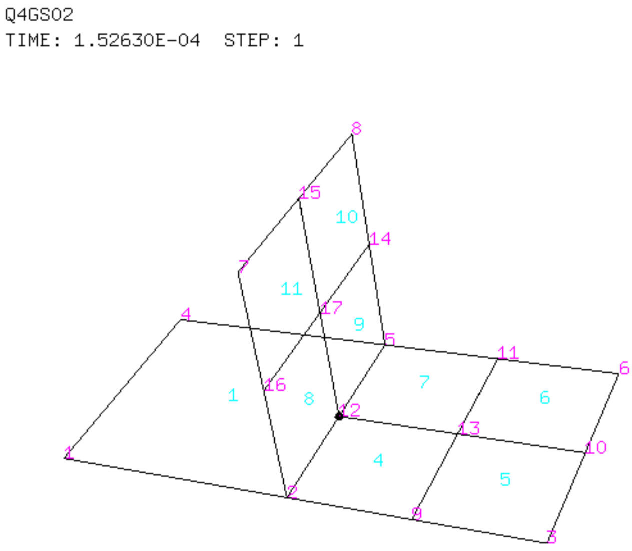
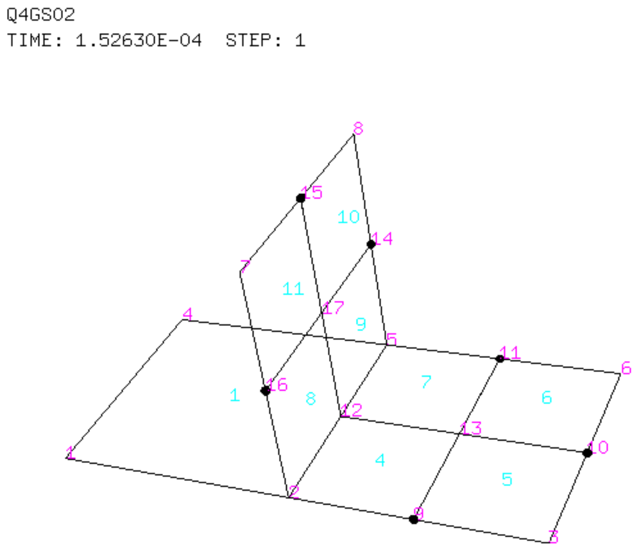


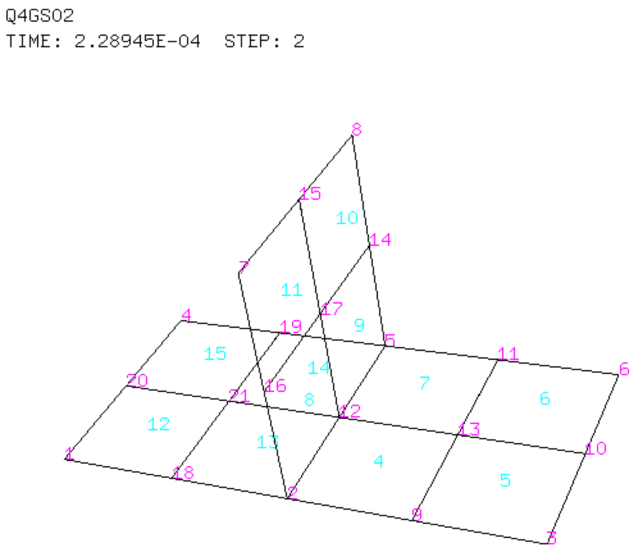
Figure 8 - Progressive mesh refinement in case Q4GS02



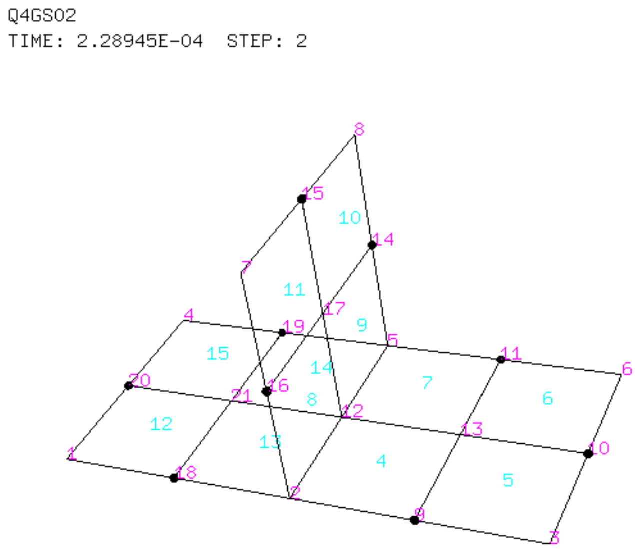
Hanging nodes after SPLI 3



B-hanging nodes after SPLI 3



Hanging nodes after SPLI 1



B-hanging nodes after SPLI 1

Figure 9 - Progressive mesh refinement in case Q4GS02 (cont'd)

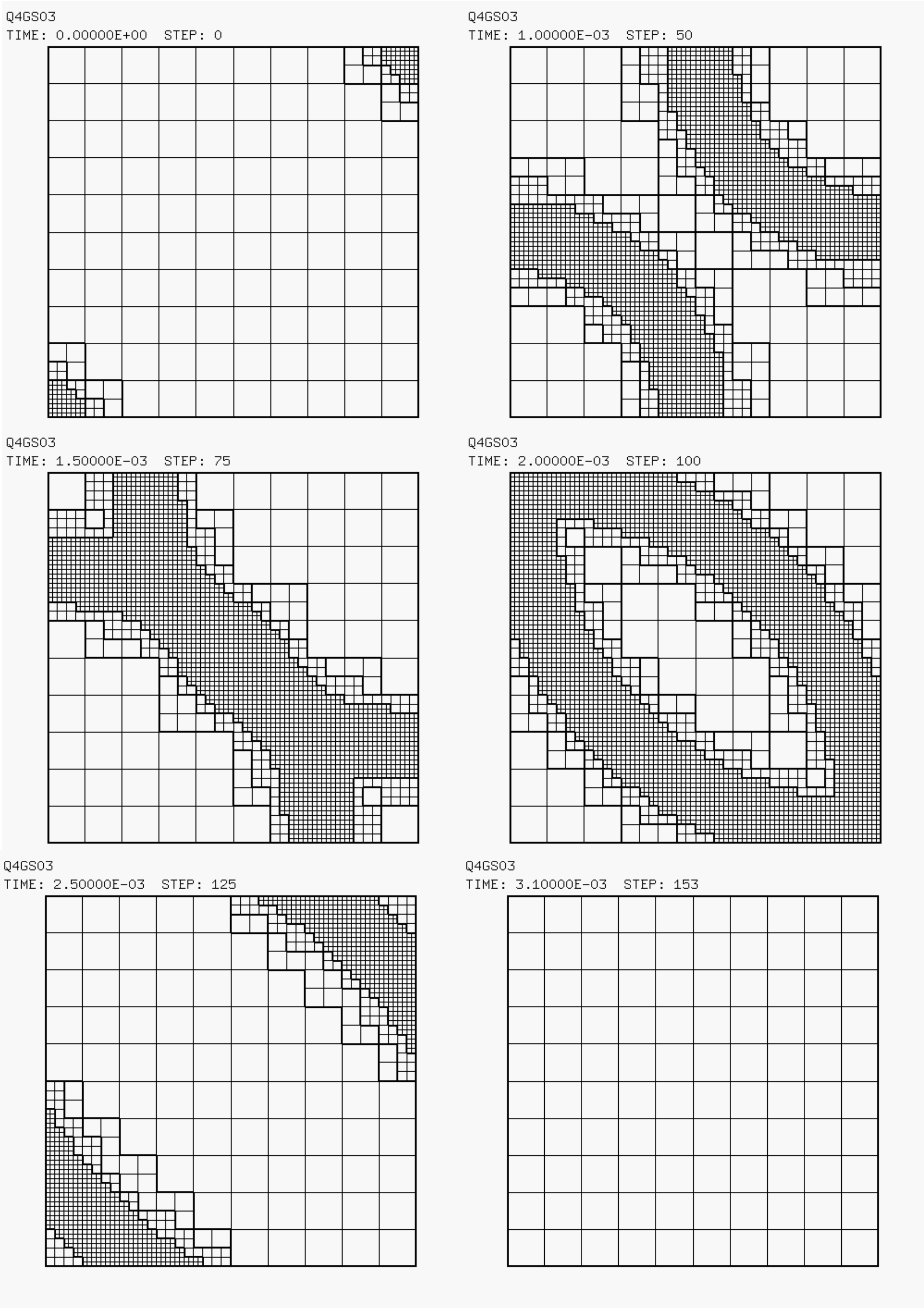


Figure 10 - Mesh adaptation in case Q4GS03

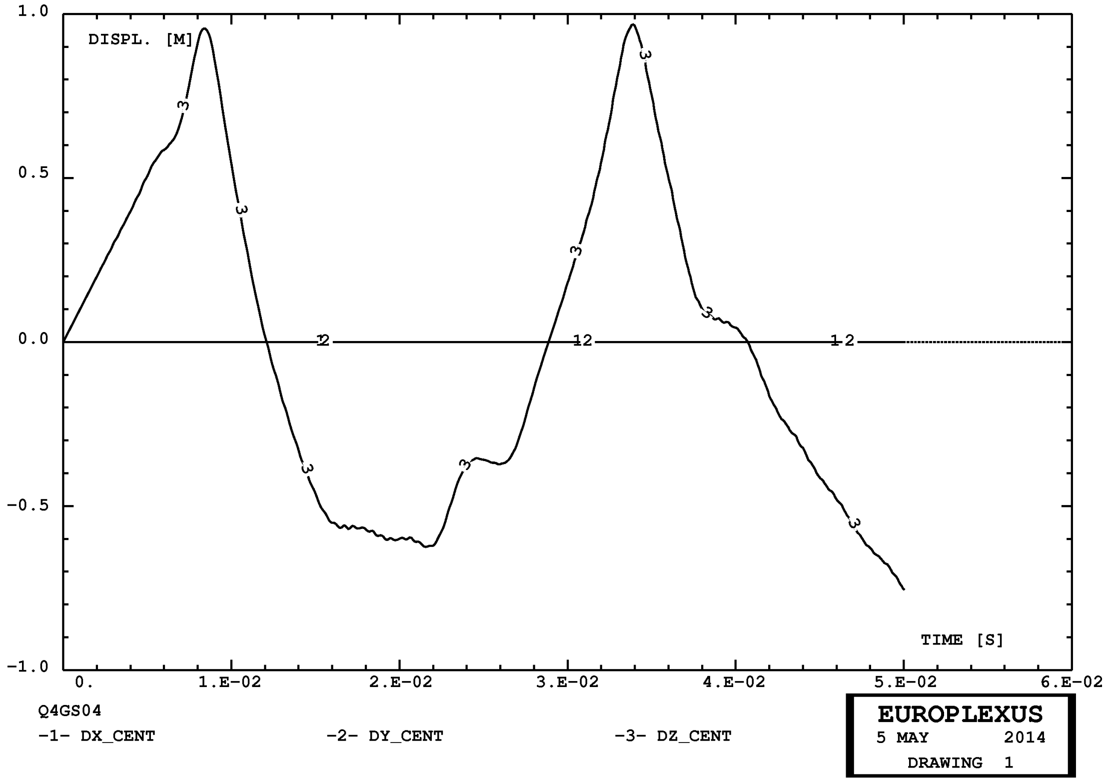


Figure 11 - Plate displacement (central node) in test Q4GS04

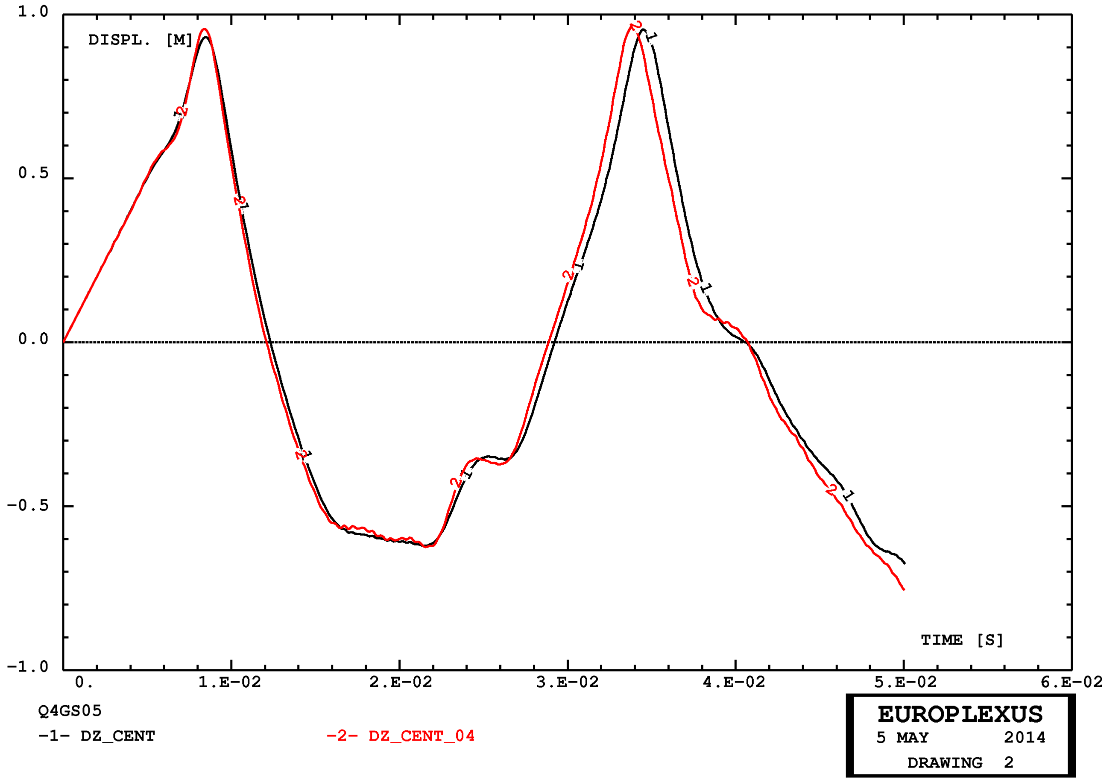


Figure 12 - Plate displacement (central node) in tests Q4GS05 and Q4GS04

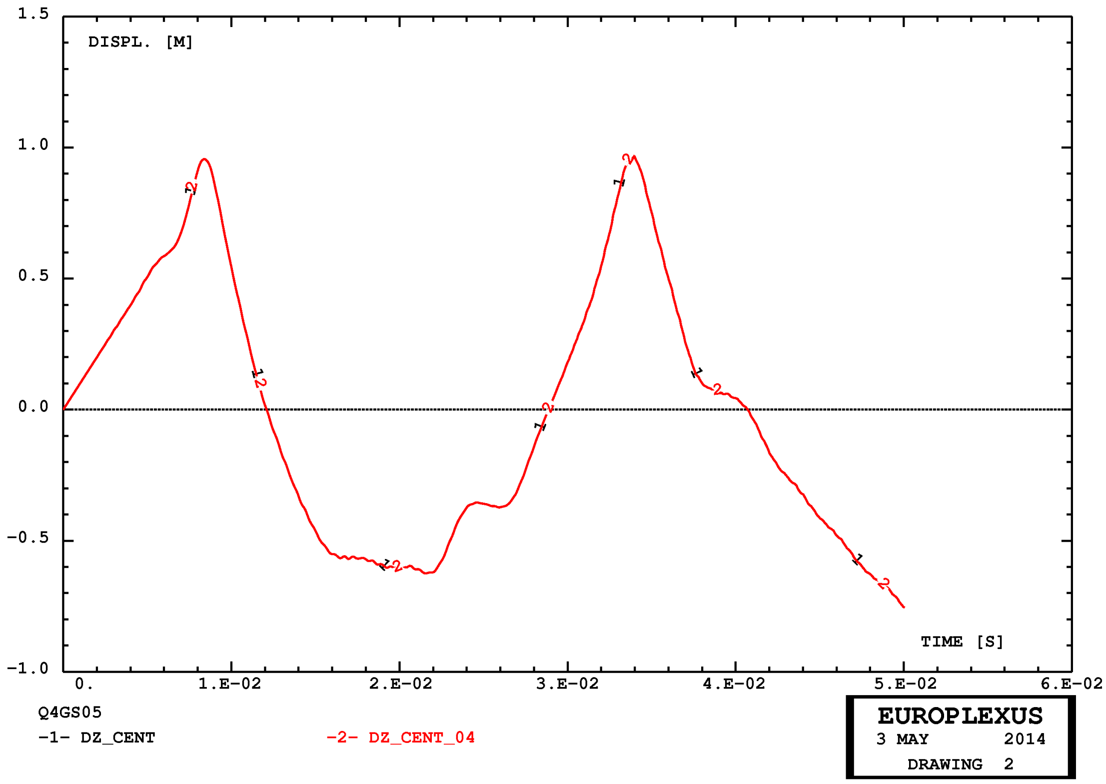


Figure 13 - Plate displacement (central node) in tests Q4GS05 and Q4GS04 (corrected)

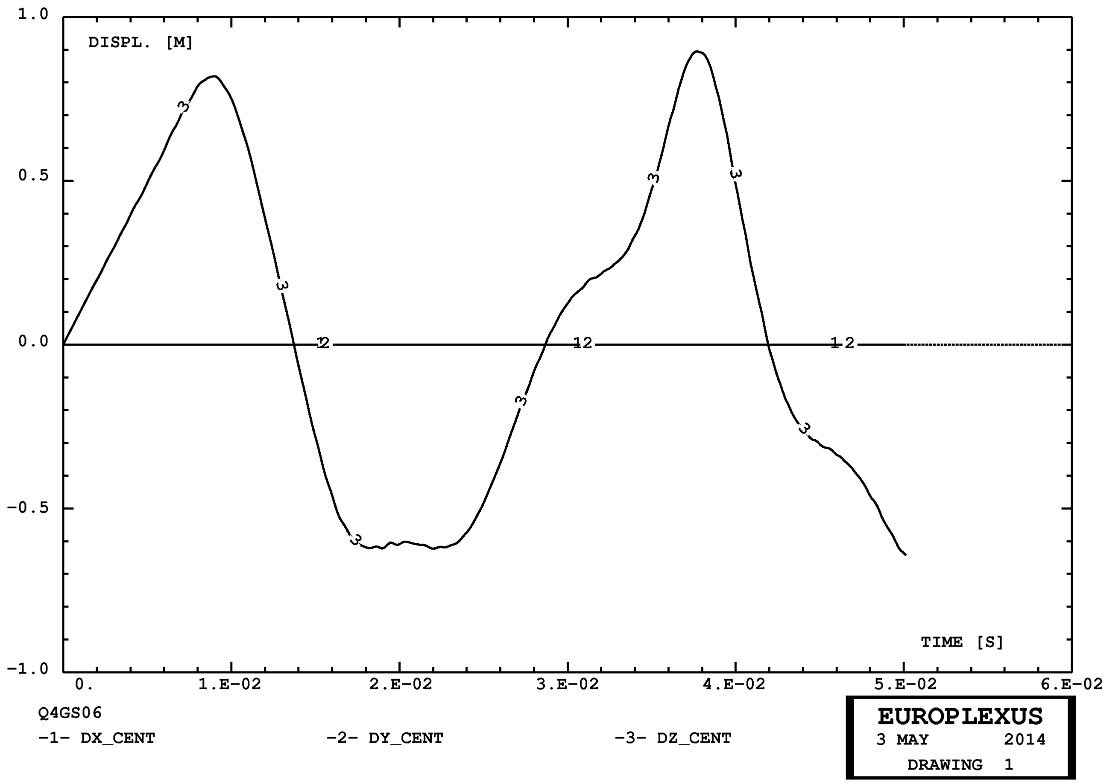


Figure 14 - Plate displacements (central node) in test Q4GS06

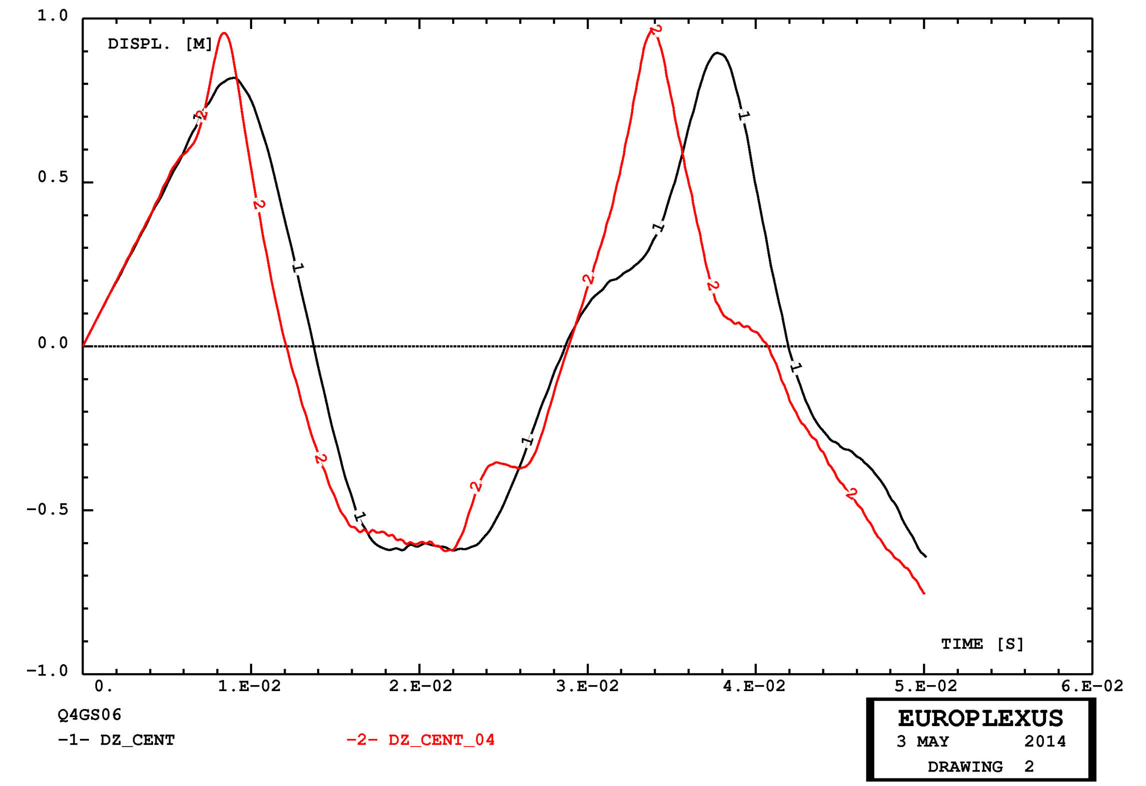


Figure 15 - Plate displacements (central node) in tests Q4GS06 and Q4GS04

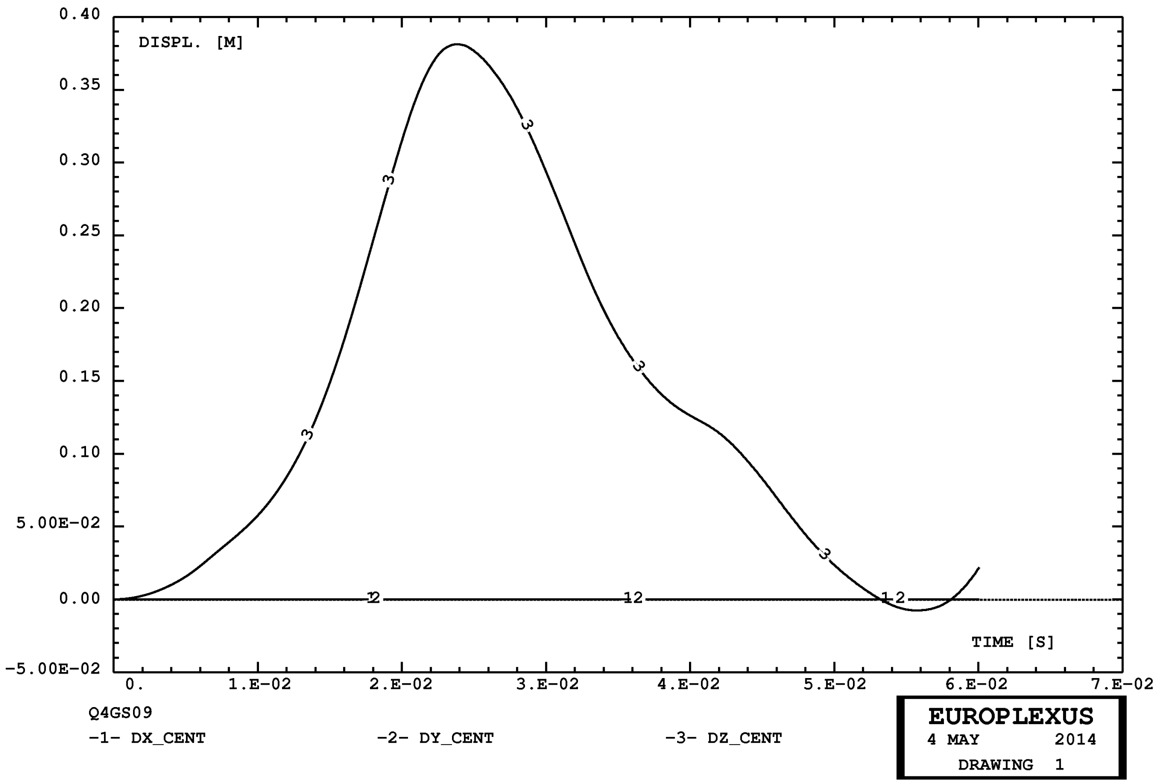


Figure 16 - Plate displacements (central node) in test Q4GS09

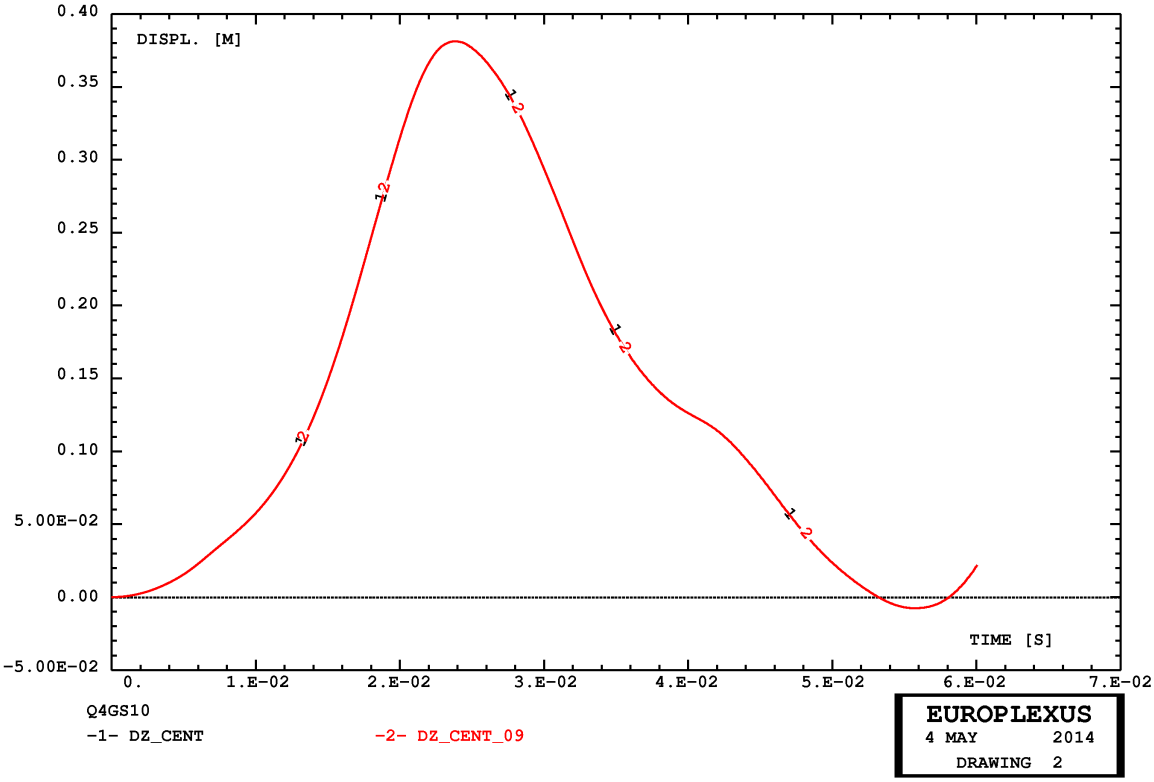


Figure 17 - Plate displacements (central node) in tests Q4GS09 and Q4GS10

3.3 Checking element splitting / unsplitting for T3GS

The next examples are similar to those presented in Section 3.1 but use the T3GS element instead of the Q4GS. The calculations are summarized in Table 3.

Case	Base Mesh	Notes	Steps	CPU [s]	Els*step
T3GS01	2 T3GS	Progressive refinement / unrefinement by ADAP SPLI / USPL	4	—	—
T3GS02	3 T3GS	Progressive refinement / unrefinement by ADAP SPLI / USPL	6	—	—
T3GS03	200 T3GS	2 WAVE directives MAXL 4 to adapt the mesh	235	5.4	759,628

Table 3 - Calculations to check element splitting / unsplitting with T3GS

T3GS01

This test is identical to Q4GS01 apart from the element used (T3GS instead of Q4GS). Results are presented in Figure 18.

T3GS02

This test is identical to Q4GS02 apart from the element used (T3GS instead of Q4GS). Results are presented in Figures 19 and 20.

T3GS03

This test is identical to Q4GS03 apart from the element used (T3GS instead of Q4GS). Results are presented in Figure 21.

3.4 Elastic oscillations of a plate with T3GS

The next examples are similar to those presented in Section 3.2 but use the T3GS element instead of the Q4GS. The calculations are summarized in Table 3.

Case	Base Mesh	Notes	Steps	CPU [s]	Els*step
T3GS09	800 T3GS	Non-adaptive solution with IMPE PIMP	1,061	2.0	1,699,200
T3GS10	200 T3GS	Adaptive solution with IMPE PIMP (INIT ADAP SPLI LEVE 2)	1,061	2.8	1,699,200

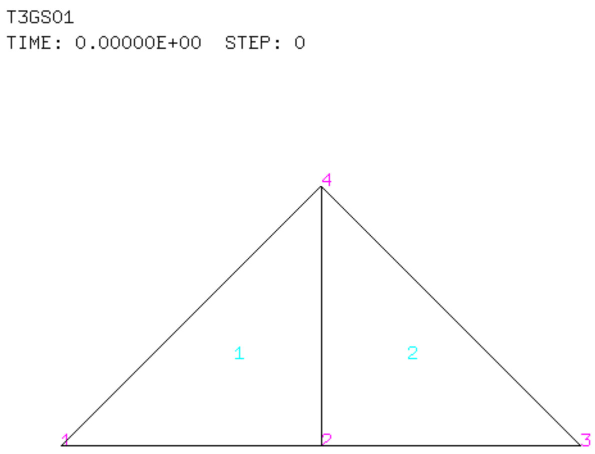
Table 4 - Calculations of the elastic oscillations of a plate with T3GS

T3GS09

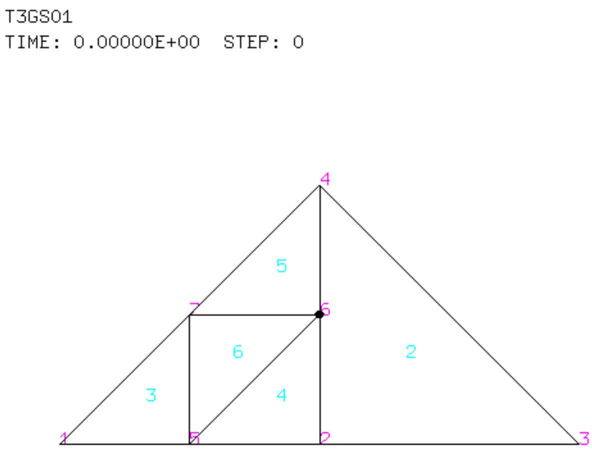
This test is identical to Q4GS09 apart from the element used (T3GS instead of Q4GS). Results are presented in Figure 22.

T3GS10

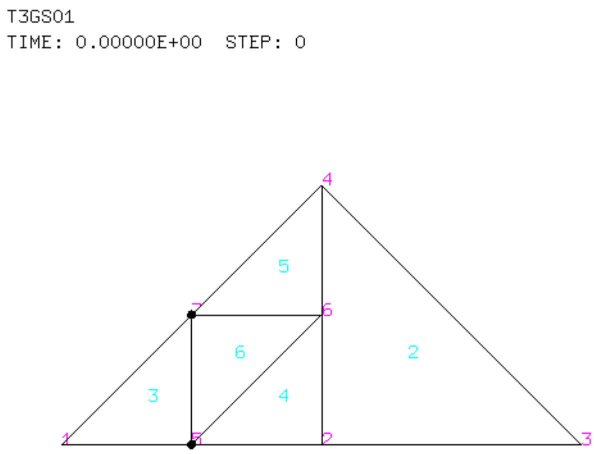
This test is similar to Q4GS10 but uses T3GS instead of Q4GS. The solution is compared in Figure 23 with the reference (T3GS09), showing perfect agreement.



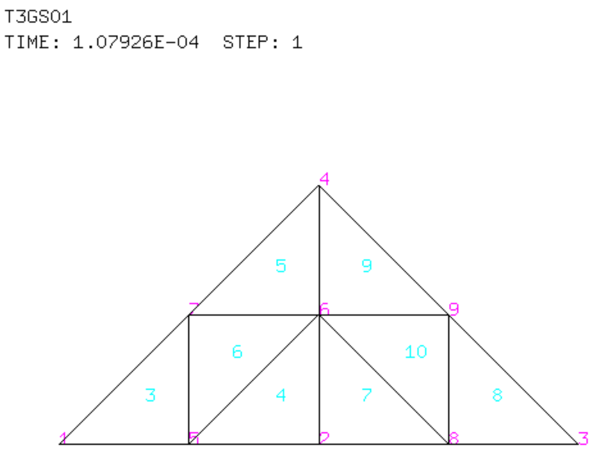
Base mesh
(no hanging
and no b-hanging
nodes)



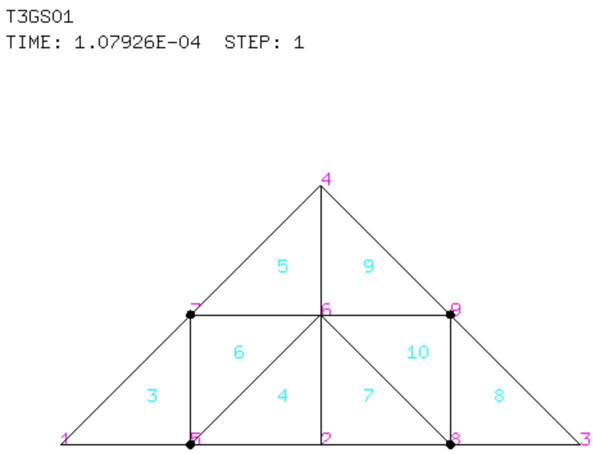
Hanging nodes after SPLI 1



B-hanging nodes after SPLI 1

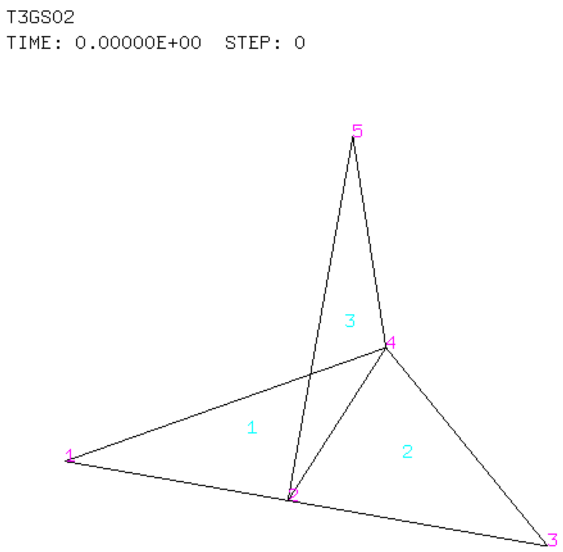


Hanging nodes after SPLI 4

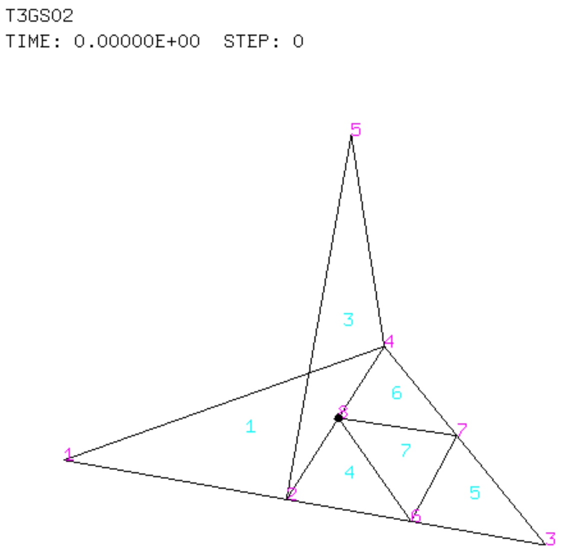


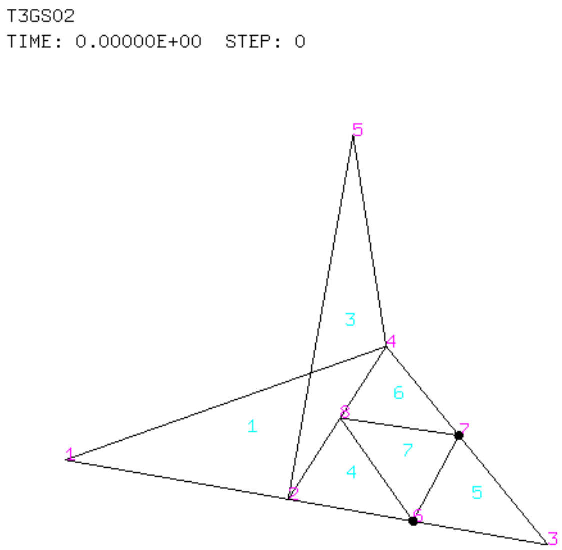
B-hanging nodes after SPLI 4

Figure 18 - Progressive mesh refinement in case T3GS01



Base mesh
(no hanging
and no b-hanging
nodes)

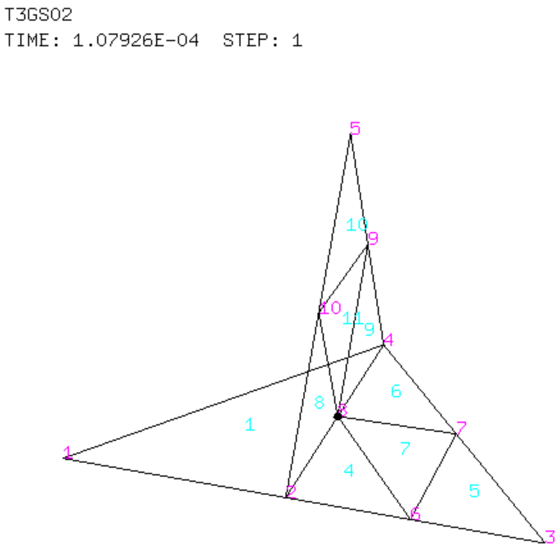




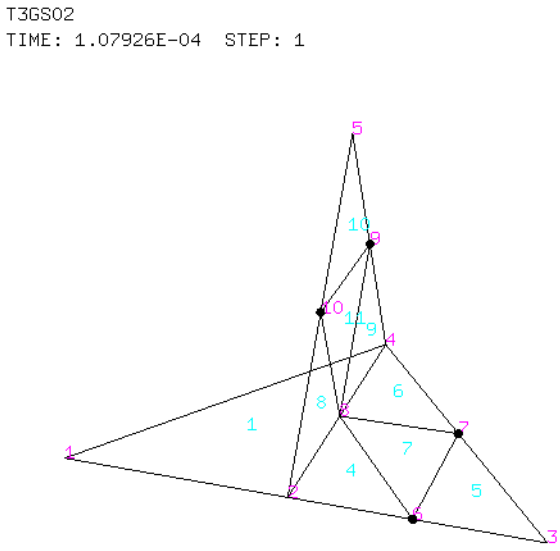
Hanging nodes after SPLI 2

B-hanging nodes after SPLI 2

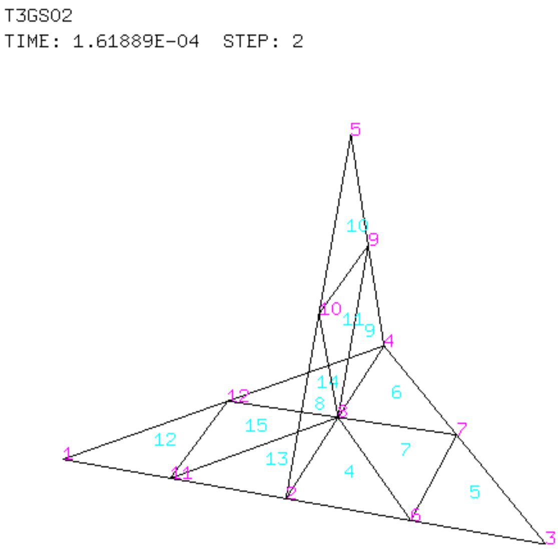
Figure 19 - Progressive mesh refinement in case T3GS02



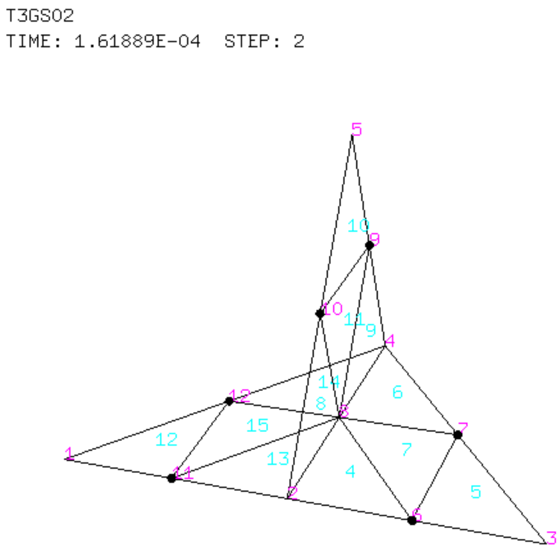
Hanging nodes after SPLI 3



B-hanging nodes after SPLI 3



Hanging nodes after SPLI 1



B-hanging nodes after SPLI 1

Figure 20 - Progressive mesh refinement in case T3GS02 (cont'd)

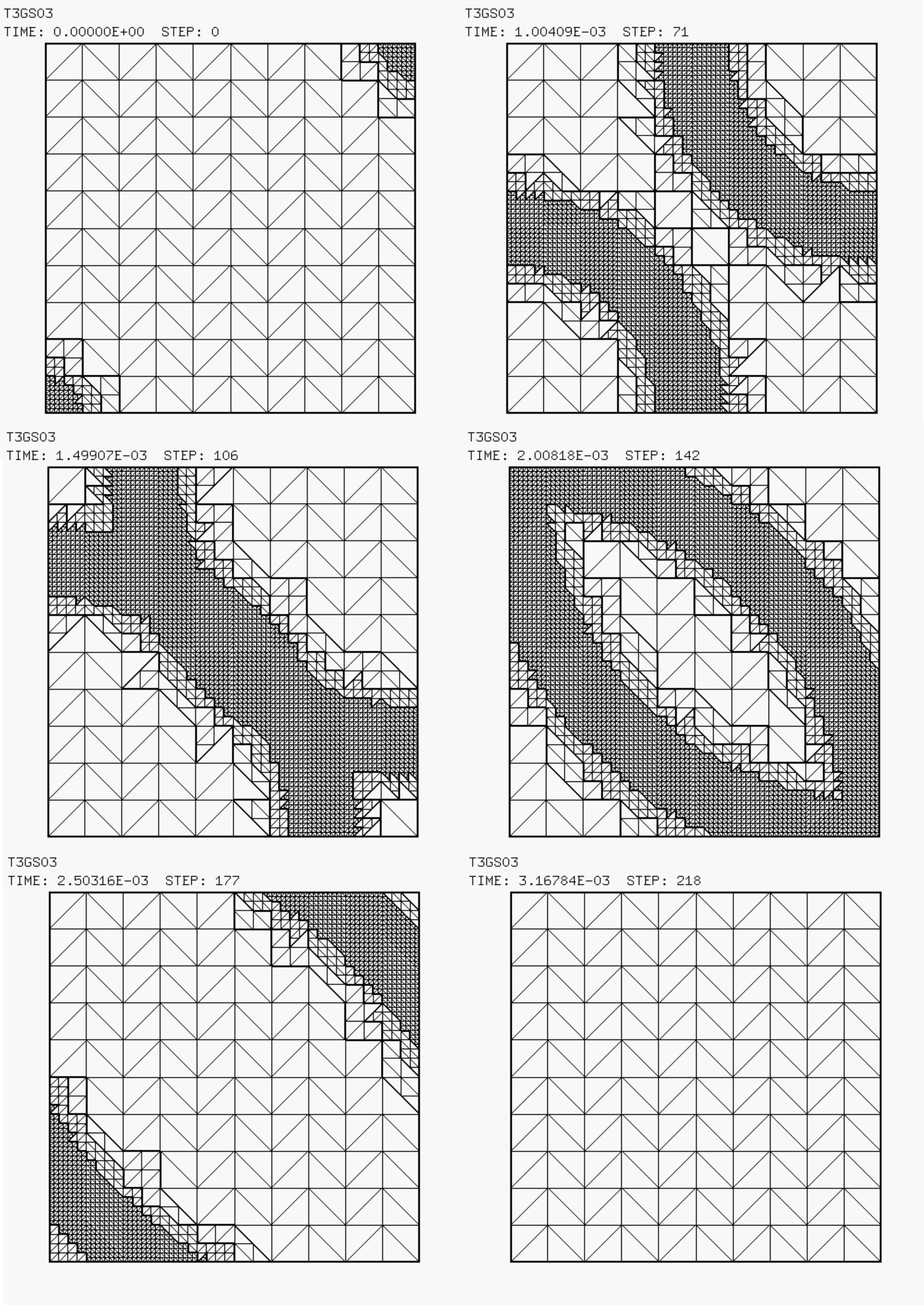


Figure 21 - Mesh adaptation in case T3GS03

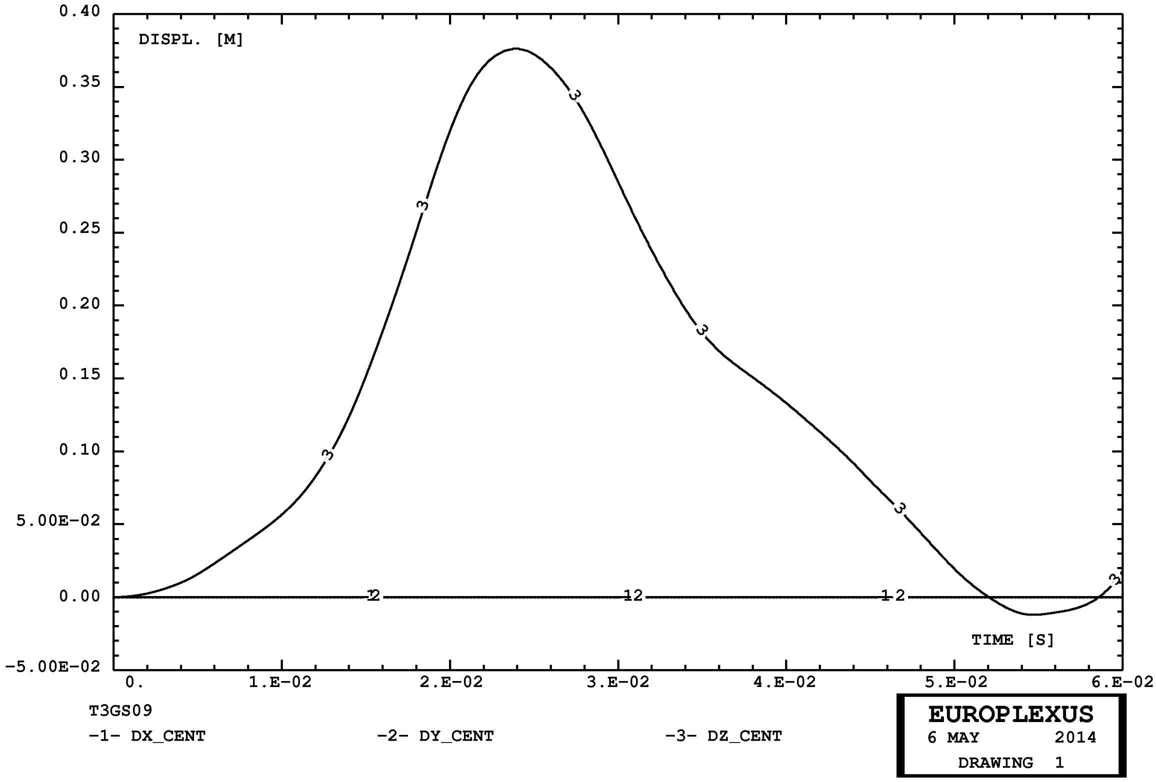


Figure 22 - Plate displacements (central node) in test T3GS09

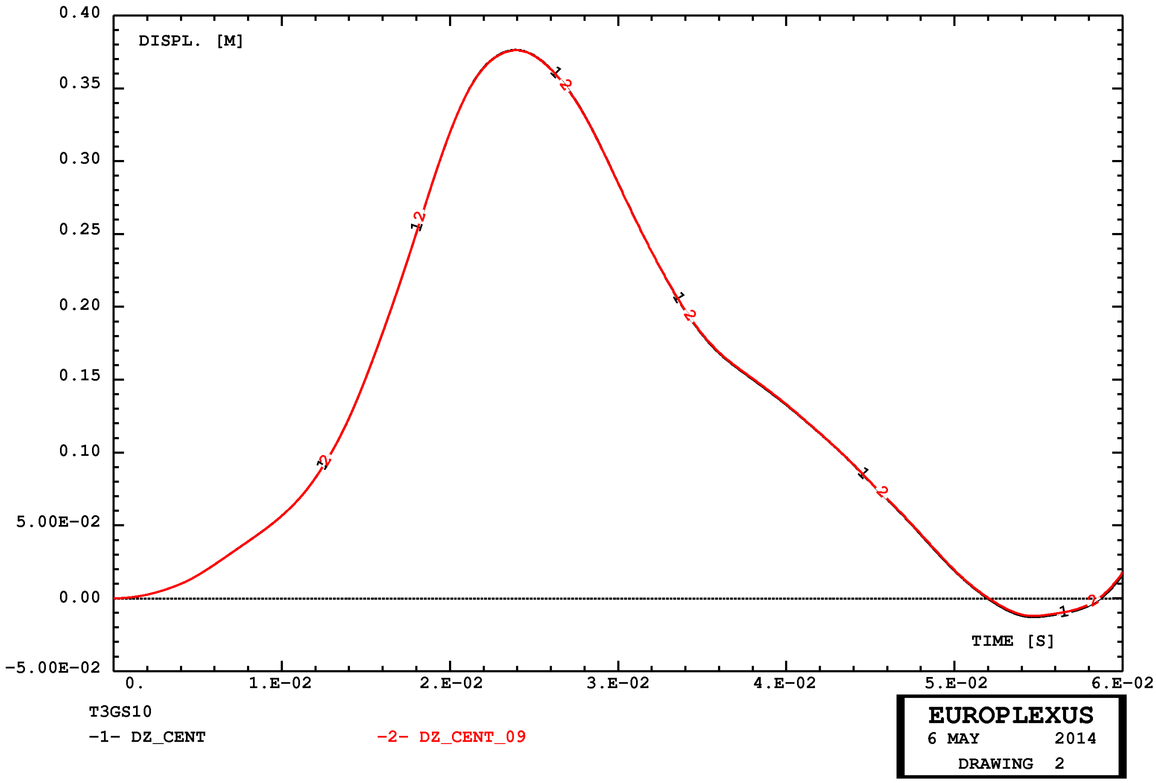


Figure 23 - Plate displacements (central node) in tests T3GS09 and T3GS10

3.5 Test problem for the FUN2

We now present some tests for SEG2-shaped elements, starting with the cable/bar element FUN2. Consider the modified (1D) Verdugo problem presented in Section 6 of reference 5: a 1D bar contains a bell-shaped initial-velocity (zero-stress) pulse which, as time advances, must split into two pulses, each one half of the initial one, and traveling in opposite directions. The calculations are summarized in Table 5.

Case	Base Mesh	Notes	Steps	CPU [s]	Els*step
VE1D17	20 FUN2	Initial refinement to level 4 by INIT ADAP. OPTI RCON	51	0.6	4,634
VE1D18	20 FUN2	Same as 17 but ADAP INDI VITE ACCE	51	0.6	4,712

Table 5 - Calculations of modified 1D Verdugo test with FUN2

VE1D17

This test is identical to VE1D16 of reference [5], apart from the dimensioning for ADAP. Results are presented in Figure 24. The solution is now symmetric with respect to the central point of the bar, but still presents some oscillations. Several programming bugs have been corrected in the implementation and a new more general strategy has been formulated which accounts for structural junctions in the error indicator (but these are not present in this particular test).

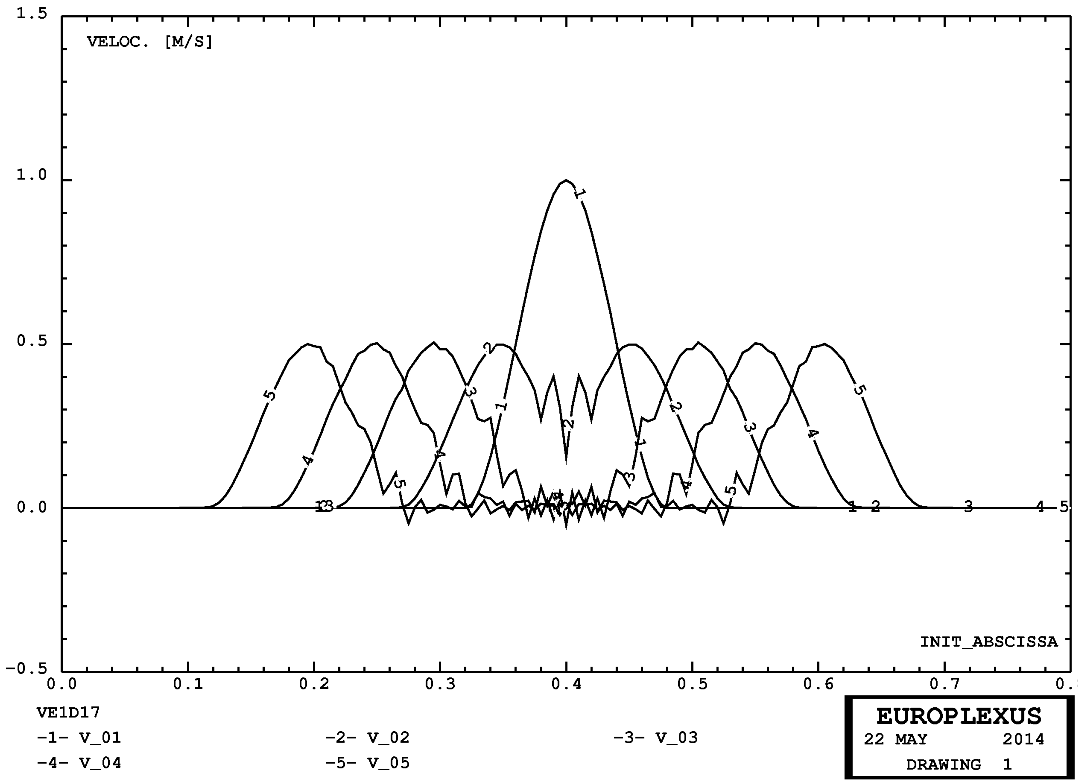


Figure 24 - X-velocity distributions along horizontal axis in test VE1D17

In order to investigate the onset of oscillations, we consider the velocity profiles and the element level profiles at the various steps, see Figures 25 to 29.

Figure 25 shows the situation at the initial time (after the initial element splitting performed by `INIT ADAP`), i.e. at step 0. The element level (which has a maximum value of 4 and a minimum value of 1) is transformed by first subtracting 1 and then dividing by 3, so that it fits in the range 0 to 1, like the velocity, and this enhances the visual representation. Due to the `RCON` option, some elements have levels intermediate between 1 and 4 already at the initial step.

At step 3, shown in Figure 26, the solution is still smooth. No unsplitting has taken place since the curvature of the velocity profile is still important over the whole pulse.

At step 10, the pulse (temporarily) assumes a flat shape (plateau) in the central region, just before splitting up in two pulses (so to say). AT this moment, the curvature goes to zero in the central region (before assuming opposite-sign non-zero values at the next step), and because of this the error indicator algorithm decides to unsplit the elements as shown in Figure 28 for step 11. At this moment the solution is still smooth. However, at the next step (step 12 shown in Figure 29) the curvature reappears (with the opposite sign), so that elements are split again. But this induces some oscillations in the solution, which then remain until the end since the material is purely elastic.

VE1D18

This test is identical to VE1D17 but it uses two indicators instead of just one: the `VITE` indicator like in the previous case and the `ACCE` indicator (this keyword is new and has been added in the code as part of the present work). Both indicators are based upon the curvature.

Since when the velocity profile is flat (see Figure 27) the acceleration profile has some non-zero curvature, it is hoped that this can solve the problem of intermittent element unsplitting and re-splitting observed in case VE1D17.

As a matter of fact, the solution becomes very smooth, as shown in Figure 30 (to be compared with Figure 24). Figures 31 and 32 show the profiles of velocity, element level and acceleration (divided by $1.\times 10^5$) at steps 11 and 12 of test VE1D18. The combined use of the two indicators avoids the intermittent un-splitting re-splitting observed in the previous case.

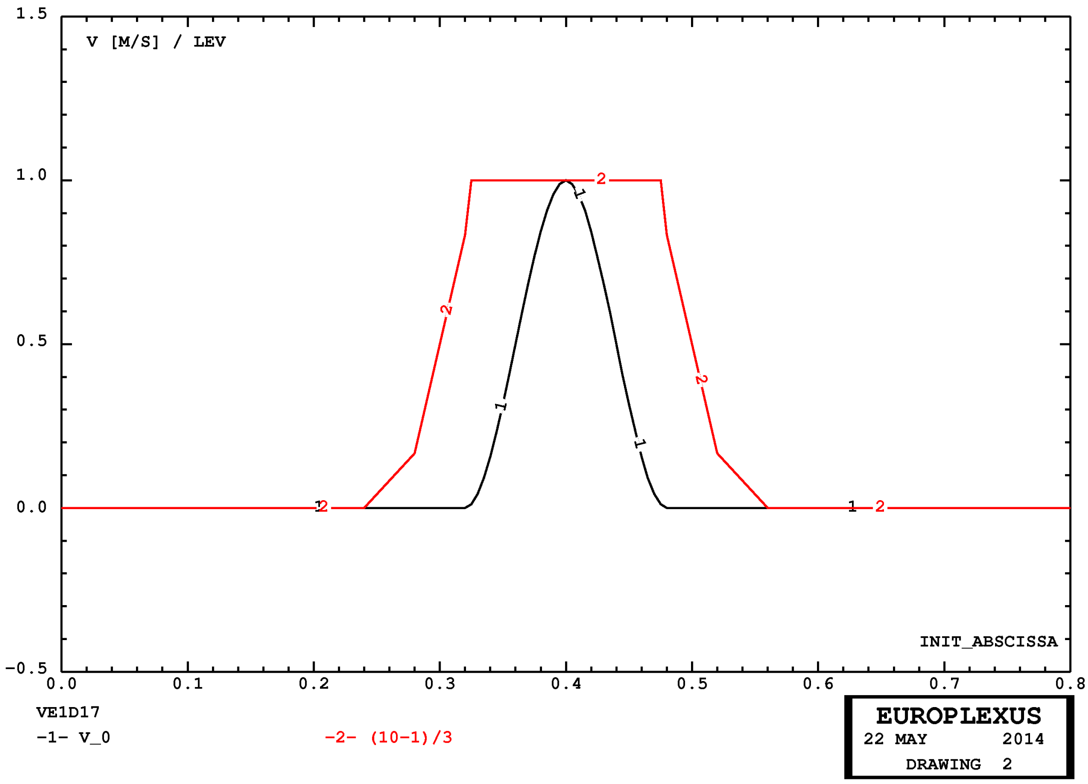


Figure 25 - X-velocity and element level profiles at step 0 in test VE1D17

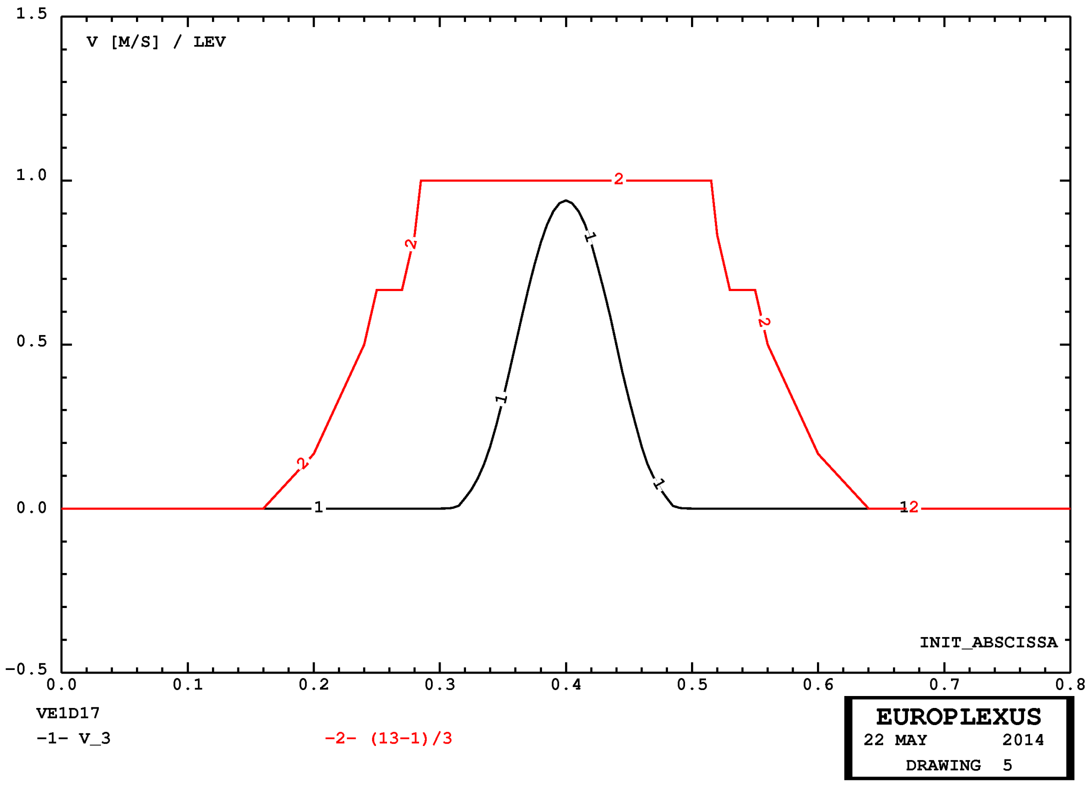


Figure 26 - X-velocity and element level profiles at step 3 in test VE1D17

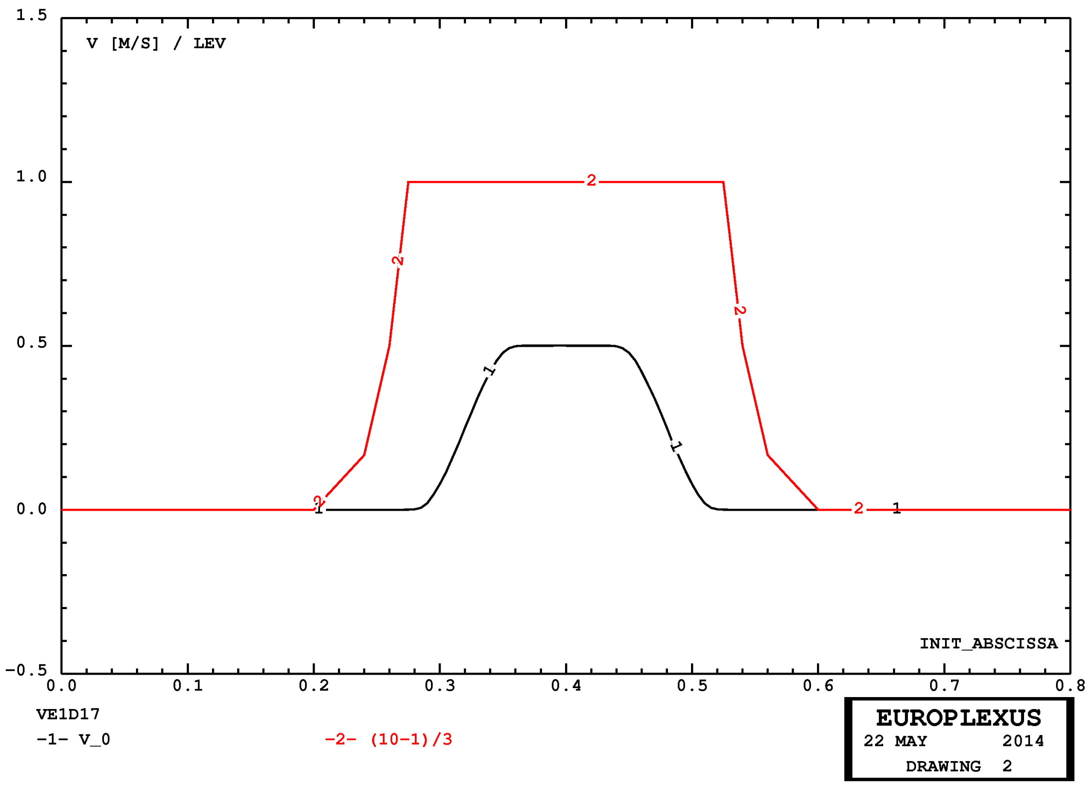


Figure 27 - X-velocity and element level profiles at step 10 in test VE1D17

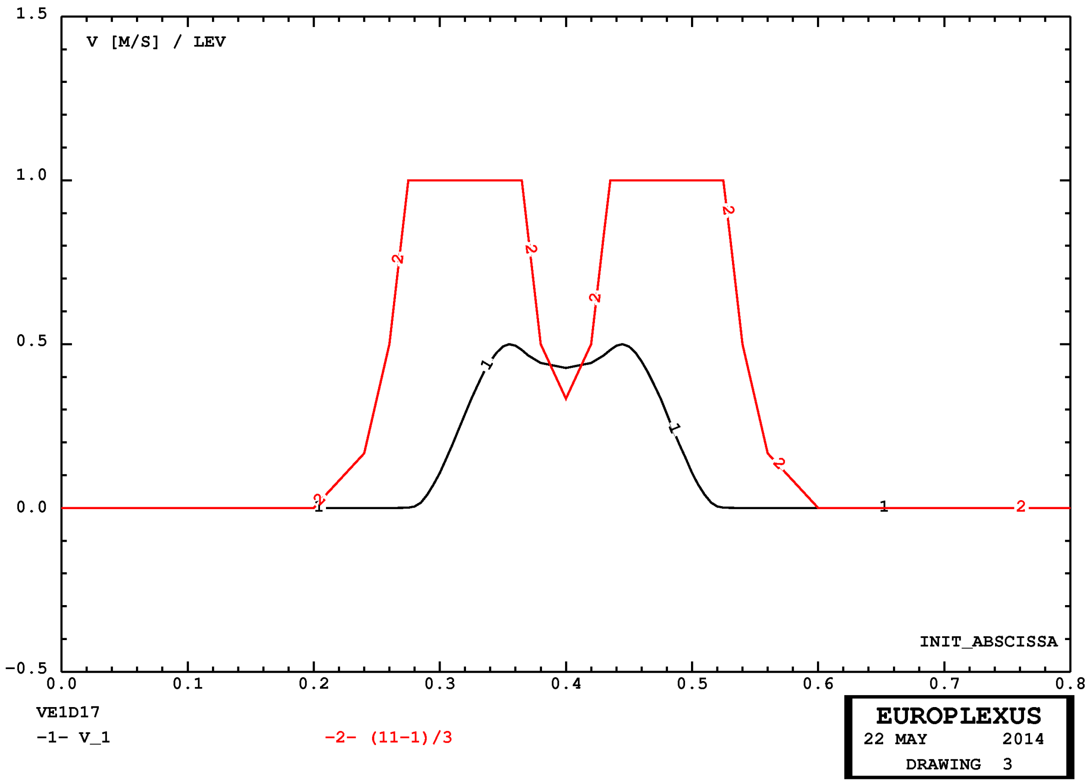


Figure 28 - X-velocity and element level profiles at step 11 in test VE1D17

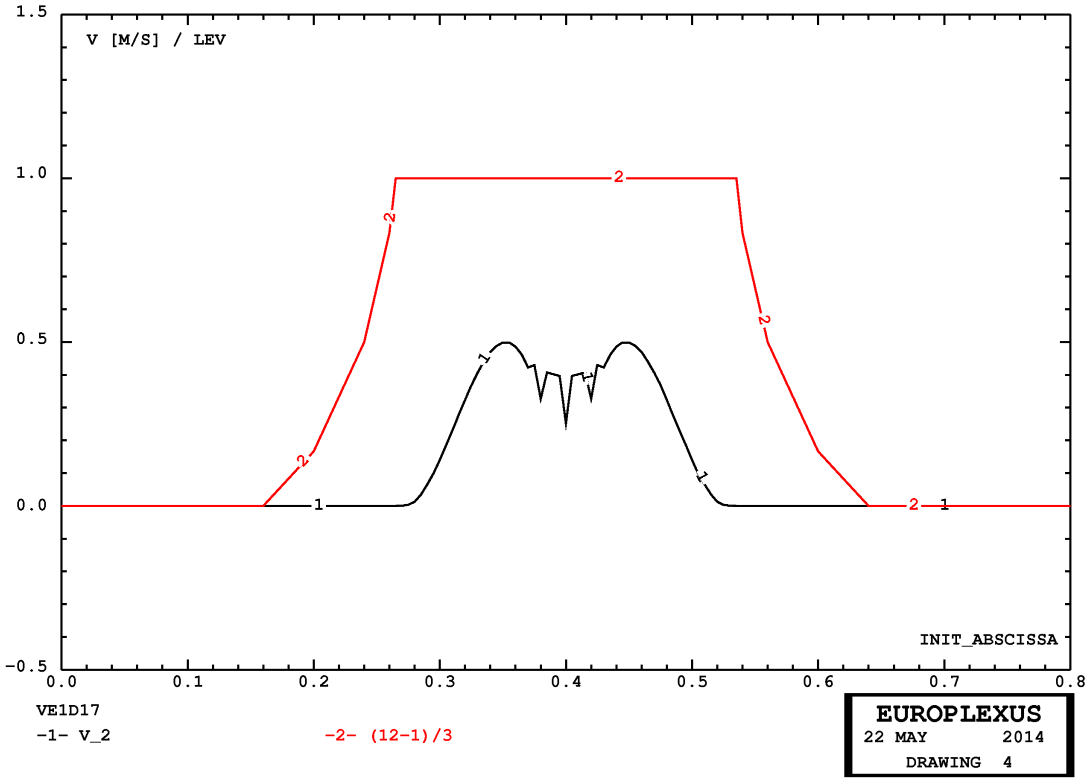


Figure 29 - X-velocity and element level profiles at step 12 in test VE1D17

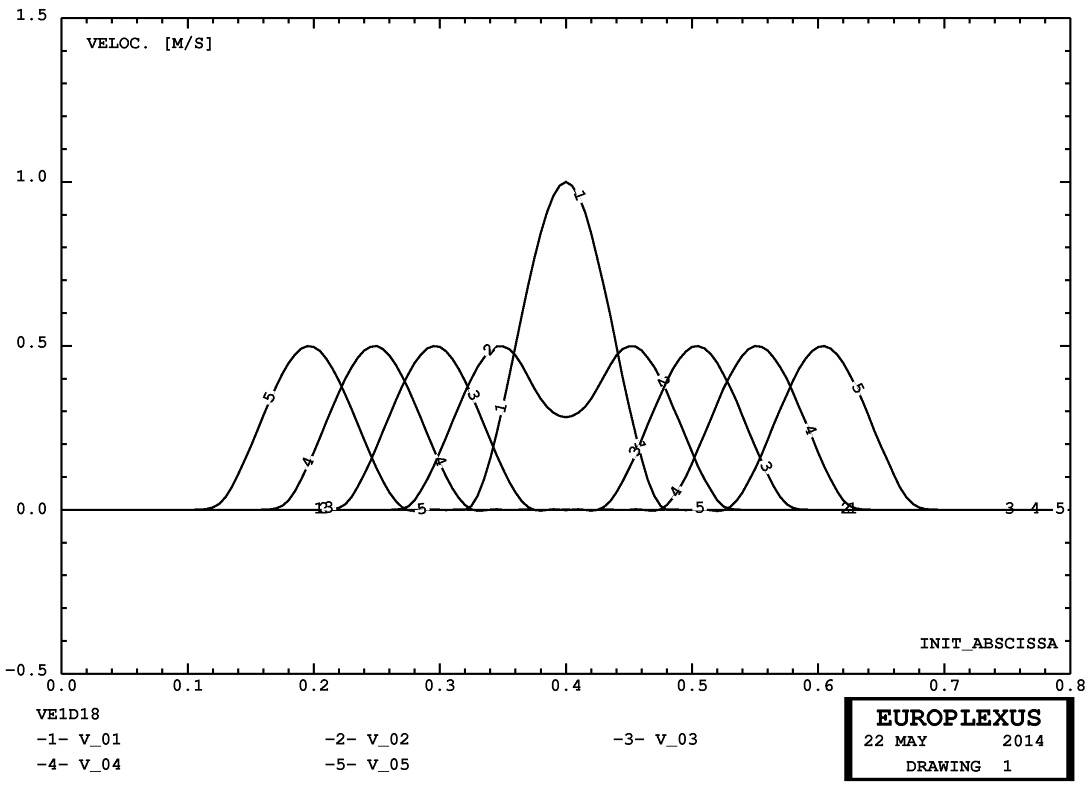


Figure 30 - X-velocity distributions along horizontal axis in test VE1D18

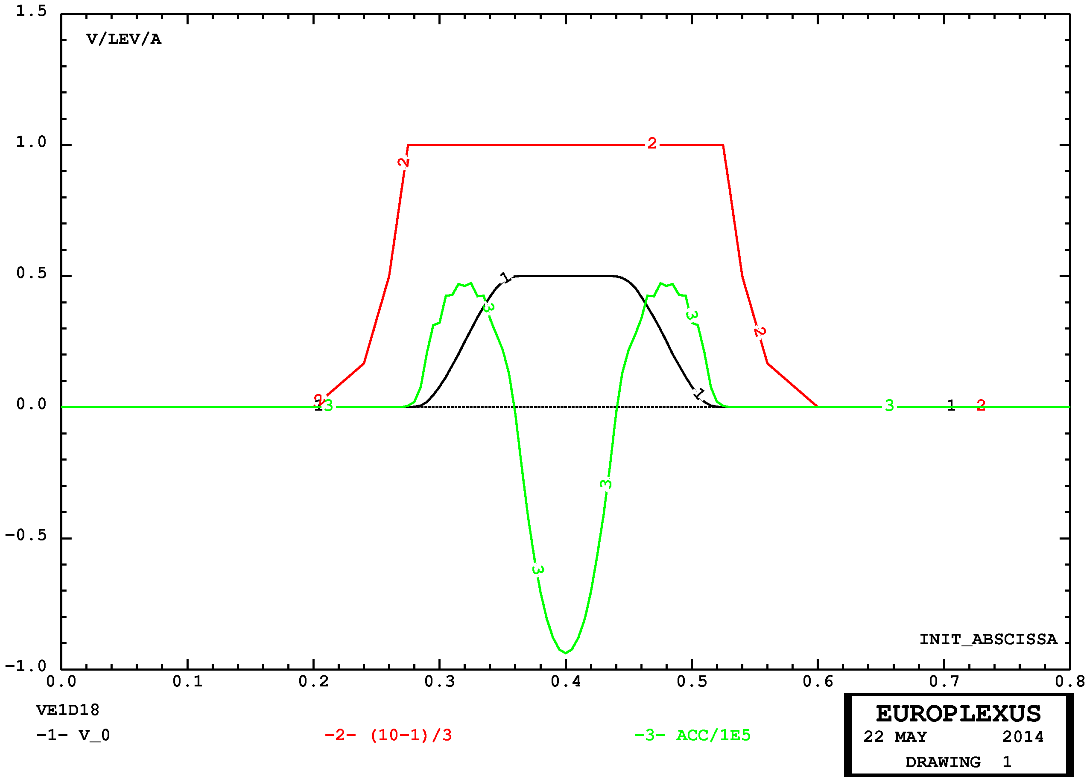


Figure 31 - Velocity, element level and acceleration profiles at step 11 in test VE1D18

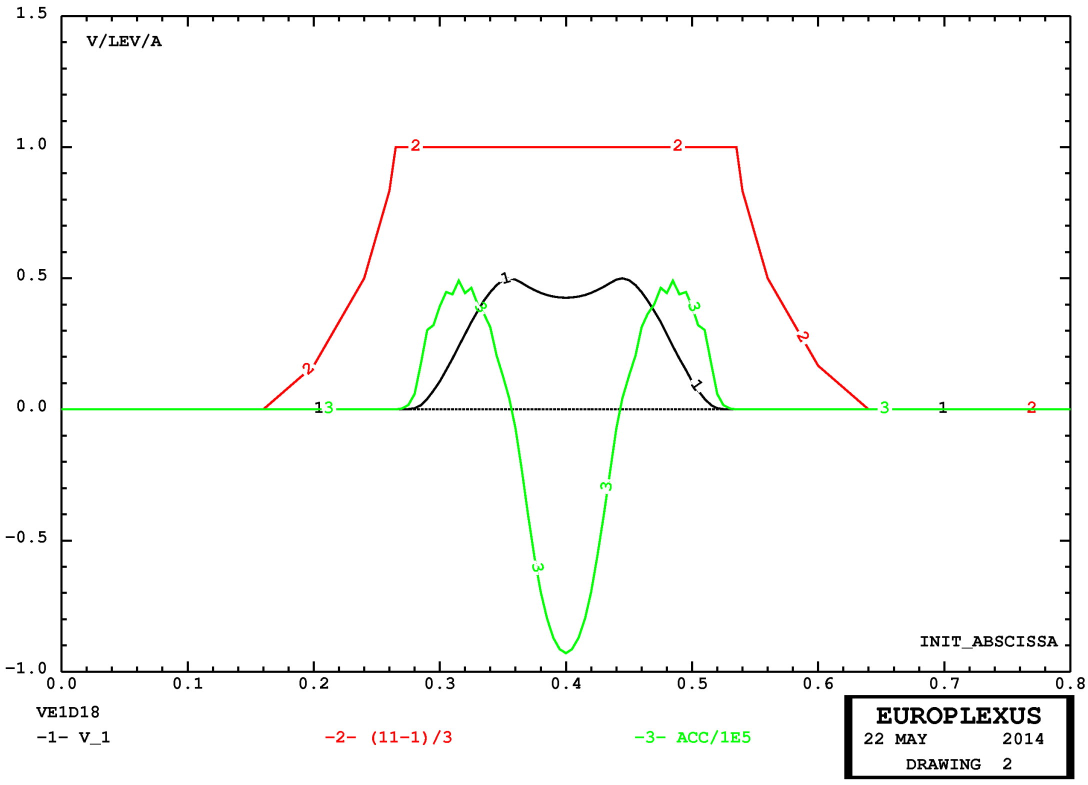


Figure 32 - Velocity, element level and acceleration profiles at step 12 in test VE1D18

3.6 Test problems for the ED01

Finally, we present some tests for the 2D shell/beam/ element ED01. A beam of length 10 m is clamped at both edges and is loaded either by an initial velocity, or by an applied pressure. The calculations are summarized in Table 5.

Case	Base Mesh	Notes	Steps	CPU [s]	Els*step
ED0101	16 ED01	Reference solution, no adaptivity, initial velocity	375	1.6	6,016
ED0102	8 ED01	Base mesh with 8 elements, INIT ADAP LEVE 2	375	4.2	6,016
ED0103	16 ED01	Reference solution, no adaptivity, applied pressure	375	1.8	12,032
ED0104	8 ED01	Base mesh with 8 elements, INIT ADAP LEVE 2	375	5.1	12,032

Table 6 - Calculations of modified 1D Verdugo test with FUN2

ED0101

This test uses no adaptivity and a mesh of 16 beam elements ED01. An initial velocity of 1 m/s is applied at all internal nodes, but the two nodes closer to the ends have a velocity of 0.5 m/s in order to comply with the subsequent (adaptive) solution.

ED0102

This test is similar to ED0101 but uses a base mesh of 8 beam elements ED01. The mesh is refined to level 2 at the initial time by INIT ADAP directive and then kept constant throughout the whole transient. An initial velocity of 1 m/s is applied at all internal base nodes, so that an initial velocity of 0.5 m/s results in the two descendent nodes close to the ends of the beam.

The solution is in excellent agreement with the reference, as shown in Figures 33 (displacements) and 34 (velocities).

ED0103

This test is identical to ED0101 (no adaptivity) but uses an applied pressure (by means of CL22 elements) instead of an initial velocity to load the beam. It is assumed as a reference for the subsequent solution.

ED0103

This test is identical to ED0102 (adaptivity) but uses an applied pressure (by means of CL22 elements) instead of an initial velocity to load the beam. The solution is in excellent agreement with the reference, as shown in Figures 33 (displacements) and 34 (velocities).

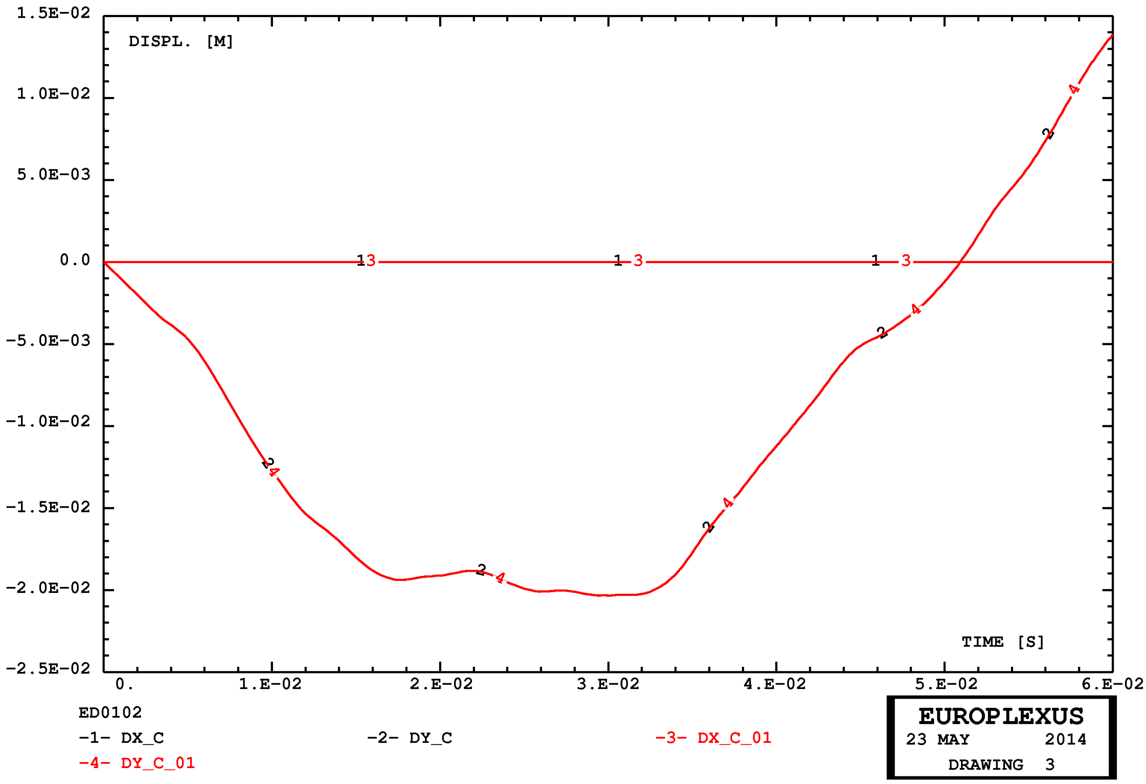


Figure 33 - Displacement of central node in tests ED0101 and ED0102

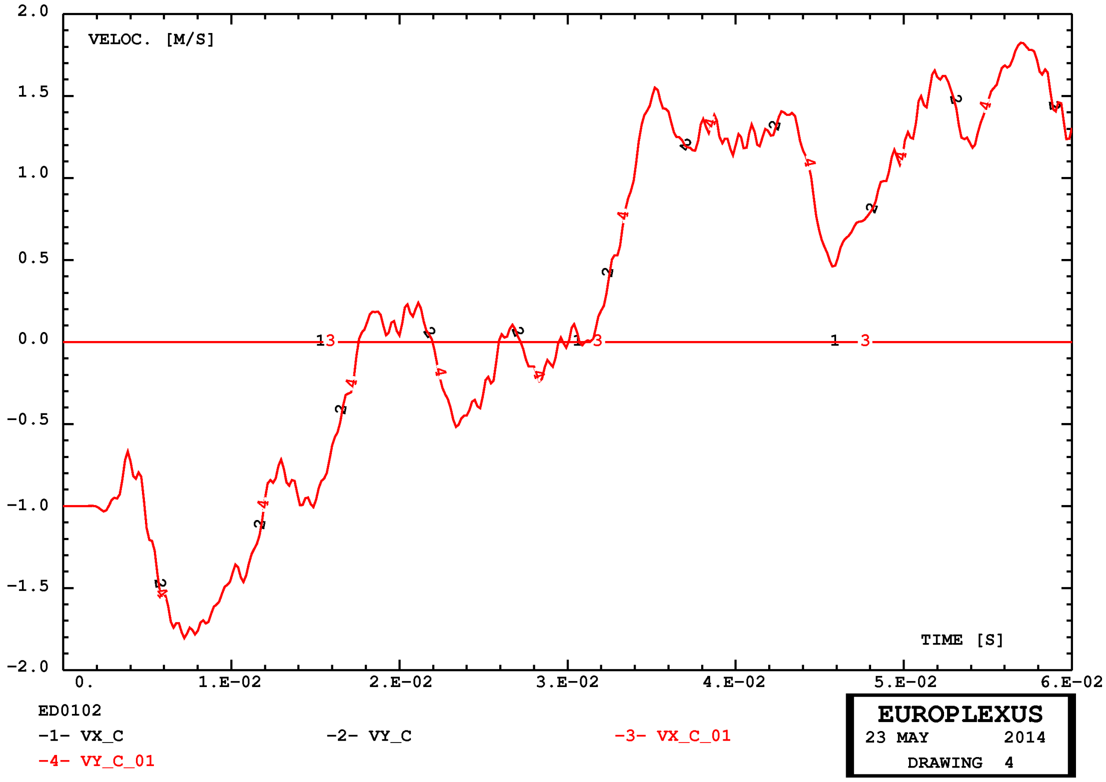


Figure 34 - Velocity of central node in tests ED0101 and ED0102

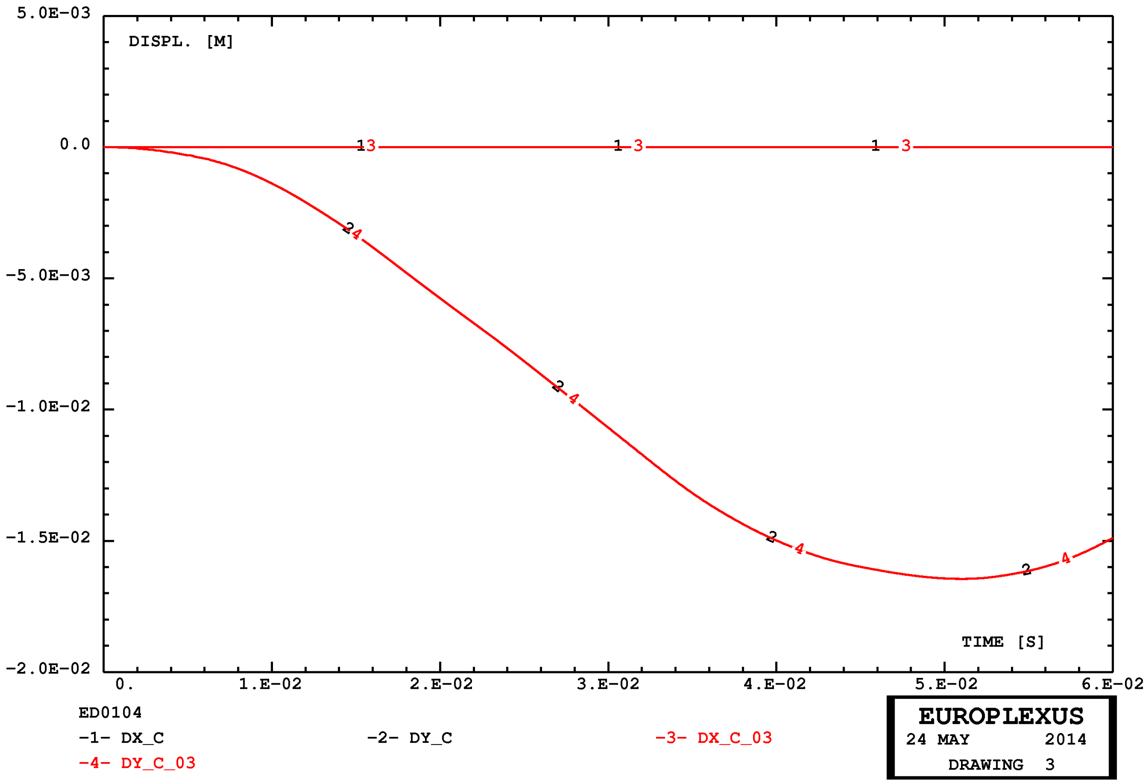


Figure 35 - Displacement of central node in tests ED0103 and ED0104

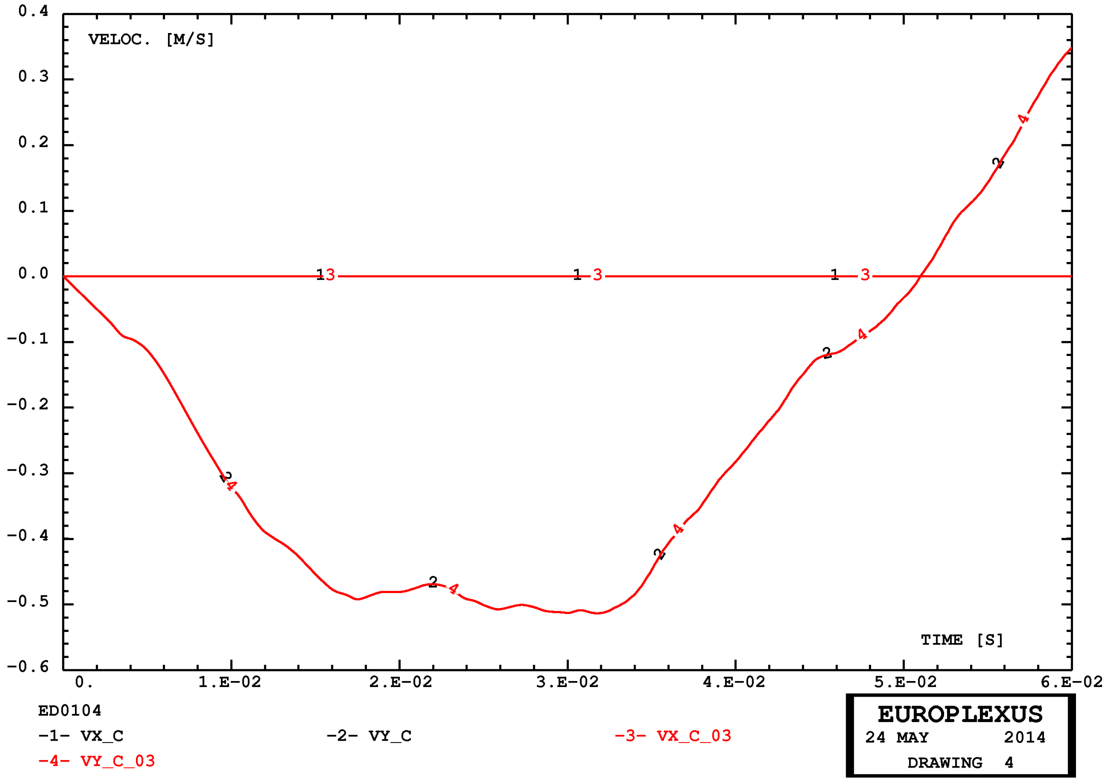


Figure 36 - Velocity of central node in tests ED0103 and ED0104

4. References

[1] F. CASADEI, P. DÍEZ, F. VERDUGO: “A Data Structure for Adaptivity in EUROPLEXUS”, JRC Technical Note PUBSY N. JRC60795, September 2010.

[2] F. CASADEI, P. DÍEZ, F. VERDUGO: “Adaptivity in FE Models for Fluids in EUROPLEXUS”, JRC Technical Note PUBSY N. JRC61622, November 2010.

[3] F. CASADEI, P. DÍEZ, F. VERDUGO: “Adaptive 3D Refinement and Un-refinement of 8-node Solid and Fluid Hexahedra in EUROPLEXUS”, JRC Technical Note PUBSY N. JRC63833, March 2011.

[4] F. CASADEI, P. DÍEZ, F. VERDUGO: “Implementation of a 2D Adaptivity Indicator for Fast Transient Dynamics in EUROPLEXUS”, JRC Technical Note PUBSY N. JRC64506, April 2011.

[5] F. CASADEI, P. DÍEZ, F. VERDUGO: “Further Development of 2D Adaptivity Error Indicators in EUROPLEXUS”, JRC Technical Note, PUBSY No. JRC66337, September 2011.

[6] F. VERDUGO, P. DÍEZ, F. CASADEI: “Natural quantities of interest in linear elastodynamics for goal oriented error estimation and adaptivity”, Proceedings of the V International Conference on Adaptive Modeling and Simulation (ADMOS 2011), D. Aubry and P. Díez (Eds), Paris, France, 6-8 June 2011.

[7] F. VERDUGO, P. DÍEZ, F. CASADEI: “General form of the natural quantities of interest for goal oriented error assessment and adaptivity in linear elastodynamics”, Submitted for publication in the International Journal for Numerical Methods in Engineering, DOI: 10.1002/nme, PUBSY No. JRC65788, July 2011.

[8] F. CASADEI, G. VALSAMOS, P. DÍEZ, F. VERDUGO: “Implementation of Adaptivity in 2D Cell Centred Finite Volumes in EUROPLEXUS”, Technical Note, PUBSY No. JRC67859, December 2011.

[9] F. CASADEI, P. DÍEZ, F. VERDUGO: “Implementation of Adaptivity in 3D Cell Centred Finite Volumes in EUROPLEXUS”, Technical Note, PUBSY No. JRC68168, December 2011.

[10] F. CASADEI, P. DÍEZ, F. VERDUGO: “Testing Adaptivity in 2D Cell Centred Finite Volumes with the CDEM Combustion Model in EUROPLEXUS”, Technical Note, PUBSY No. JRC68333, December 2011.

[11] F. CASADEI, P. DÍEZ, F. VERDUGO: “An algorithm for mesh refinement and un-refinement in fast transient dynamics”, International Journal of Computational Methods, DOI 10.1142/S0219876213500187, Vol. 10, No. 4, pp. 1350018-1 / 1350018-31, 2013.

[12] F. CASADEI, G. VALSAMOS, A. BECCANTINI: “Combination of Mesh Adaptivity with Fluid-

Structure Interaction in EUROPLEXUS", Technical Note, in publication, 2014.

[13] F. CASADEI, M. LARCHER, G. VALSAMOS: "Adaptivity with Simplex Elements in EUROPLEXUS", Technical Note, in publication, 2014.

[14] F. CASADEI, M. LARCHER, G. VALSAMOS: "Adaptivity in CEA's Fluid Elements in EUROPLEXUS", Technical Note, in publication, 2014.

[15] "*EUROPLEXUS User's Manual*", on-line version: http://europlexus.jrc.ec.europa.eu.

Appendix

Sample input files

This Section contains, in alphabetical file order, the listings of all input files related to the examples which were proposed in the previous Sections.

adja01.epx

```
ADJA01
ECHO
 CONV WIN
LAGR TRID
opti dump
GEOM LIBR POIN 8 PRIS 1 Q4GS 3 CL3D 1 TERM
 0 0 0  1 0 0  2 0 0
 0 1 0  1 1 0  2 1 0
 1 0 1  1 1 1
 1 7 2 4 8 5
 1 2 5 4
 2 3 6 5
 2 5 8 7
 2 3 6 5
COMP GROU 3 'dumm' LECT 1 TERM
            'stru' LECT 2 PAS 1 4 TERM
            'clxx' LECT 5 TERM
     EPAI 1.0 LECT stru TERM
     COUL VERT LECT stru TERM
          JAUN LECT clxx TERM
          ROSE LECT dumm TERM
MATE LINE RO 8000. YOUN 2.D11 NU 0.3
          LECT stru TERM
     IMPE ABSO RO 8000 C 5000 LECT clxx TERM
     FANT 0.001 LECT dumm TERM
ECRI DEPL VITE ACCE FINT FEXT FLIA FDEC CONT ECRO FREQ 1
     FICH ALIC FREQ 1
OPTI NOTE LOG 1
CALC TINI 0. TEND 1.0 NMAX 0
FIN
```

adja02.epx

```
ADJA02
ECHO
 CONV WIN
LAGR DPLA
opti dump
GEOM LIBR POIN 5 ED01 4 CL2D 1 TERM
 0 1   1 1   1 0
 1 2   2 2
 1 2
 3 2
 2 4
 4 5
 2 3
COMP GROU 2 'stru' LECT 1 PAS 1 4 TERM
            'clim' LECT 5 TERM
     EPAI 0.1 LECT stru TERM
     COUL VERT LECT stru TERM
          JAUN LECT clim TERM
MATE VM23 RO 8000. YOUN 2.D11 NU 0.3 ELAS 2.D11
          TRAC 1 2.D11 1
          LECT stru TERM
     IMPE ABSO RO 8000 C 5000 LECT clim TERM
ECRI DEPL VITE ACCE FINT FEXT FLIA FDEC CONT ECRO FREQ 1
     FICH ALIC FREQ 1
OPTI NOTE LOG 1
CALC TINI 0. TEND 1.0 NMAX 0
FIN
```

ed0101.epx

```
ED0101
ECHO
!CONV win
LAGR CPLA
GEOM LIBR POIN 17 ED01 16 TERM
 0 0  1 0  2 0  3 0  4 0  5 0  6 0  7 0  8 0  9 0  10 0
 11 0  12 0  13 0  14 0  15 0  16 0
 1 2
 2 3
 3 4
 4 5
 5 6
 6 7
 7 8
 8 9
 9 10
 10 11
 11 12
 12 13
 13 14
 14 15
 15 16
 16 17
COMP GROU 1 'mesh' LECT 1 PAS 1 16 TERM
     COUL VERT LECT mesh TERM
     EPAI 5.E-1 LECT mesh TERM
MATE VM23 RO 8000. YOUN 2.E11 NU 0.0 ELAS 2.E11
          TRAC 1 2.E11 1.D0
          LECT mesh TERM
LINK COUP BLOQ 123 LECT 1 17 TERM
INIT VITE 2 -0.5 LECT 2       16 TERM
     VITE 2 -1.0 LECT 3 PAS 1 15 TERM
ECRI COOR DEPL VITE ACCE FINT FEXT FLIA CONT ECRO FREQ 1
     FICH ALIC FREQ 1
OPTI NOTE LOG 1 DPMA
  LNKS STAT
CALC TINI 0. TEND 60.0E-3
SUIT
ED0101A
ECHO
RESU ALIC GARD PSCR
SORT GRAP
PERF 'ed0101a.pun'
AXTE 1.0 'Time [s]'
COUR 1 'dx_c' DEPL COMP 1 NOEU LECT 9 TERM
COUR 2 'dy_c' DEPL COMP 2 NOEU LECT 9 TERM
COUR 3 'vx_c' VITE COMP 1 NOEU LECT 9 TERM
COUR 4 'vy_c' VITE COMP 2 NOEU LECT 9 TERM
TRAC  1  2  AXES 1.0 'DISPL. [M]'
TRAC  3  4  AXES 1.0 'VELOC. [M/S]'
LIST  1  2  AXES 1.0 'VELOC. [M/S]'
LIST  3  4  AXES 1.0 'VELOC. [M/S]'
*==============================================================
FIN
```

ed0102.epx

```
ED0102
ECHO
!CONV win
LAGR CPLA
DIME ADAP NPOI 8 ED01 16 ENDA TERM
GEOM LIBR POIN 9 ED01 8 TERM
 0 0  2 0  4 0  6 0  8 0  10 0  12 0  14 0  16 0
 1 2
 2 3
 3 4
 4 5
 5 6
 6 7
 7 8
 8 9
COMP GROU 1 'mesh' LECT 1 PAS 1 8 TERM
     COUL VERT LECT mesh TERM
     EPAI 5.E-1 LECT mesh _ed01 TERM
MATE VM23 RO 8000. YOUN 2.E11 NU 0.0 ELAS 2.E11
          TRAC 1 2.E11 1.D0
          LECT mesh _ed01 TERM
LINK COUP BLOQ 123 LECT 1 9 TERM
INIT VITE 2 -1.0 LECT 2 PAS 1 8 TERM
     ADAP SPLI LEVE 2 LECT mesh TERM
ECRI COOR DEPL VITE ACCE FINT FEXT FLIA CONT ECRO FREQ 1
     FICH ALIC FREQ 1
OPTI NOTE LOG 1 DPMA
  ADAP STAT DUMP
  LNKS STAT
CALC TINI 0. TEND 60.0E-3
SUIT
*--------------------------------------------------------------------
ED0102A
ECHO
RESU ALIC GARD PSCR
SORT GRAP
PERF 'ed0102a.pun'
AXTE 1.0 'Time [s]'
COUR 1 'dx_c' DEPL COMP 1 NOEU LECT 5 TERM
COUR 2 'dy_c' DEPL COMP 2 NOEU LECT 5 TERM
COUR 3 'vx_c' VITE COMP 1 NOEU LECT 5 TERM
COUR 4 'vy_c' VITE COMP 2 NOEU LECT 5 TERM
TRAC  1  2  AXES 1.0 'DISPL. [M]'
TRAC  3  4  AXES 1.0 'VELOC. [M/S]'
LIST  1  2  AXES 1.0 'VELOC. [M/S]'
LIST  3  4  AXES 1.0 'VELOC. [M/S]'
RCOU 11 'dx_c' FICH 'ed0101a.pun' RENA 'dx_c_01'
RCOU 12 'dy_c' FICH 'ed0101a.pun' RENA 'dy_c_01'
RCOU 13 'vx_c' FICH 'ed0101a.pun' RENA 'vx_c_01'
RCOU 14 'vy_c' FICH 'ed0101a.pun' RENA 'vy_c_01'
TRAC  1  2  11 12 AXES 1.0 'DISPL. [M]'
COLO NOIR NOIR ROUG ROUG
TRAC  3  4  13 14 AXES 1.0 'VELOC. [M/S]'
COLO NOIR NOIR ROUG ROUG
*====================================================================
FIN
```

ed0103.epx

```
ED0103
ECHO
!CONV win
LAGR CPLA
GEOM LIBR POIN 17 ED01 16 CL22 16 TERM
 0 0  1 0  2 0  3 0  4 0  5 0  6 0  7 0  8 0  9 0  10 0
 11 0  12 0  13 0  14 0  15 0  16 0
 1 2
 2 3
 3 4
 4 5
 5 6
 6 7
 7 8
 8 9
 9 10
 10 11
 11 12
 12 13
 13 14
 14 15
 15 16
 16 17
 1 2
 2 3
 3 4
```

```
 4 5
 5 6
 6 7
 7 8
 8 9
 9 10
 10 11
 11 12
 12 13
 13 14
 14 15
 15 16
 16 17
COMP GROU 2  'bar' LECT 1 PAS 1 16 TERM
            'pimp' LECT 17 PAS 1 32 TERM
     COUL VERT LECT bar TERM
          JAUN LECT pimp TERM
     EPAI 5.E-1 LECT bar TERM
MATE VM23 RO 8000. YOUN 2.E11 NU 0.0 ELAS 2.E11
          TRAC 1 2.E11 1.D0
          LECT bar TERM
     IMPE PIMP RO 8000. PRES 1.E5 PREF 0 FONC 1
          LECT pimp TERM
FONC NUM 1 TABL 2 0 1 100 1
LINK COUP BLOQ 123 LECT 1 17 TERM
ECRI COOR DEPL VITE ACCE FINT FEXT FLIA CONT ECRO FREQ 1
     FICH ALIC FREQ 1
OPTI NOTE LOG 1 DPMA
  LNKS STAT
CALC TINI 0. TEND 60.0E-3
SUIT
ED0103A
ECHO
RESU ALIC GARD PSCR
SORT GRAP
PERF 'ed0103a.pun'
AXTE 1.0 'Time [s]'
COUR 1 'dx_c' DEPL COMP 1 NOEU LECT 9 TERM
COUR 2 'dy_c' DEPL COMP 2 NOEU LECT 9 TERM
COUR 3 'vx_c' VITE COMP 1 NOEU LECT 9 TERM
COUR 4 'vy_c' VITE COMP 2 NOEU LECT 9 TERM
TRAC  1  2  AXES 1.0 'DISPL. [M]'
TRAC  3  4  AXES 1.0 'VELOC. [M/S]'
LIST  1  2  AXES 1.0 'VELOC. [M/S]'
LIST  3  4  AXES 1.0 'VELOC. [M/S]'
QUAL DEPL COMP 1 LECT 9 TERM REFE  0.00000E+0 TOLE 1.E-2
     DEPL COMP 2 LECT 9 TERM REFE -1.48898E-2 TOLE 1.E-2
     VITE COMP 1 LECT 9 TERM REFE  0.00000E+0 TOLE 1.E-2
     VITE COMP 2 LECT 9 TERM REFE  3.49512E-1 TOLE 1.E-2
*=================================================================
FIN
```

ed0104.epx

```
ED0104
ECHO
!CONV win
LAGR CPLA
DIME ADAP NPOI 8 ED01 16 CL22 16 ENDA TERM
GEOM LIBR POIN 9 ED01 8 CL22 8 TERM
 0 0  2 0  4 0  6 0  8 0  10 0  12 0  14 0  16 0
 1 2
 2 3
 3 4
 4 5
 5 6
 6 7
 7 8
 8 9
 1 2
 2 3
 3 4
 4 5
 5 6
 6 7
 7 8
 8 9
COMP GROU 2  'bar' LECT 1 PAS 1 8 TERM
            'pimp' LECT 9 PAS 1 16 TERM
     COUL VERT LECT bar TERM
          JAUN LECT pimp TERM
     EPAI 5.E-1 LECT bar _ed01 TERM
MATE VM23 RO 8000. YOUN 2.E11 NU 0.0 ELAS 2.E11
          TRAC 1 2.E11 1.D0
          LECT bar _ed01 TERM
     IMPE PIMP RO 8000. PRES 1.E5 PREF 0 FONC 1
          LECT pimp _cl22 TERM
FONC NUM 1 TABL 2 0 1 100 1
LINK COUP BLOQ 123 LECT 1 9 TERM
INIT ADAP SPLI LEVE 2 LECT bar TERM
ECRI COOR DEPL VITE ACCE FINT FEXT FLIA CONT ECRO FREQ 1
     FICH ALIC FREQ 1
OPTI NOTE LOG 1 DPMA
  ADAP STAT DUMP
  LNKS STAT
CALC TINI 0. TEND 60.0E-3
SUIT
*----------------------------------------------------------------------
ED0104A
ECHO
RESU ALIC GARD PSCR
SORT GRAP
PERF 'ed0104a.pun'
AXTE 1.0 'Time [s]'
COUR 1 'dx_c' DEPL COMP 1 NOEU LECT 5 TERM
COUR 2 'dy_c' DEPL COMP 2 NOEU LECT 5 TERM
COUR 3 'vx_c' VITE COMP 1 NOEU LECT 5 TERM
COUR 4 'vy_c' VITE COMP 2 NOEU LECT 5 TERM
TRAC  1  2  AXES 1.0 'DISPL. [M]'
TRAC  3  4  AXES 1.0 'VELOC. [M/S]'
LIST  1  2  AXES 1.0 'VELOC. [M/S]'
LIST  3  4  AXES 1.0 'VELOC. [M/S]'
RCOU 11 'dx_c' FICH 'ed0103a.pun' RENA 'dx_c_03'
RCOU 12 'dy_c' FICH 'ed0103a.pun' RENA 'dy_c_03'
RCOU 13 'vx_c' FICH 'ed0103a.pun' RENA 'vx_c_03'
RCOU 14 'vy_c' FICH 'ed0103a.pun' RENA 'vy_c_03'
TRAC  1  2  11 12 AXES 1.0 'DISPL. [M]'
COLO NOIR NOIR ROUG ROUG
TRAC  3  4  13 14 AXES 1.0 'VELOC. [M/S]'
COLO NOIR NOIR ROUG ROUG
QUAL DEPL COMP 1 LECT 5 TERM REFE  0.00000E+0 TOLE 1.E-2
     DEPL COMP 2 LECT 5 TERM REFE -1.48898E-2 TOLE 1.E-2
     VITE COMP 1 LECT 5 TERM REFE  0.00000E+0 TOLE 1.E-2
     VITE COMP 2 LECT 5 TERM REFE  3.49512E-1 TOLE 1.E-2
*=====================================================================
FIN
```

posp01.epx

```
POSP01
ECHO
 CONV WIN
LAGR TRID
opti dump
GEOM LIBR POIN 12 CUBE 1 Q4GS 1 TERM
 0 0 0  1 0 0  1 1 0  0 1 0
 0 0 1  1 0 1  1 1 1  0 1 1
 2 0 0  3 0 0
 3 1 0  2 1 0
 1 2 3 4 5 6 7 8
 9 10 11 12
COMP GROU 3 'cont' LECT 1 TERM
            'shel' LECT 2 TERM
            'stru' LECT cont shel TERM
     EPAI 1.0 LECT shel TERM
     COUL VERT LECT cont TERM
          ROSE LECT shel TERM
MATE LINE RO 8000. YOUN 2.D11 NU 0.3
          LECT stru TERM
ECRI DEPL VITE ACCE FINT FEXT FLIA FDEC CONT ECRO FREQ 1
     FICH ALIC FREQ 1
OPTI NOTE LOG 1
CALC TINI 0. TEND 1.0 NMAX 0
FIN
```

posp02.epx

```
POSP02
ECHO
 CONV WIN
LAGR TRID
opti dump
GEOM LIBR POIN 10 CUBE 1 Q4GS 1 TERM
 0 0 0  1 0 0  2 0 0
 0 1 0  1 1 0  2 1 0
 0 0 1  1 0 1
 0 1 1  1 1 1
 1 2 5 4 7 8 10 9
 2 3 6 5
COMP GROU 3 'cont' LECT 1 TERM
            'shel' LECT 2 TERM
            'stru' LECT cont shel TERM
     EPAI 1.0 LECT shel TERM
     COUL VERT LECT cont TERM
          ROSE LECT shel TERM
MATE LINE RO 8000. YOUN 2.D11 NU 0.3
          LECT stru TERM
ECRI DEPL VITE ACCE FINT FEXT FLIA FDEC CONT ECRO FREQ 1
     FICH ALIC FREQ 1
OPTI NOTE LOG 1
CALC TINI 0. TEND 1.0 NMAX 0
FIN
```

q41l01.epx

```
Q41L01
ECHO
 CONV WIN
LAGR CPLA
DIME ADAP NPOI 5 Q41L 4 ENDA TERM
opti dump dpma
GEOM LIBR POIN 6 Q41L 2 TERM
 0 0   1 0   2 0
 0 1   1 1   2 1
 1 2 5 4
 2 3 6 5
COMP GROU 1 'stru' LECT 1 PAS 1 2 TERM
     EPAI 1.0 LECT stru _q41l TERM
     COUL VERT LECT stru TERM
MATE VM23 RO 8000. YOUN 2.D11 NU 0.3 ELAS 2.D11
          TRAC 1 2.D11 1.D0
          LECT stru _q41l TERM
LINK COUP ! mandatory for adaptivity
     BLOQ 1 LECT 1 4 TERM
     BLOQ 2 LECT 1 2 3 TERM
ECRI DEPL VITE ACCE FINT FEXT FLIA FDEC CONT ECRO FREQ 1
     FICH ALIC FREQ 1
OPTI NOTE LOG 1
     ADAP DUMP STAT CHEC
     LNKS DUMP STAT DIAG
CALC TINI 0. TEND 1.0 NMAX 1
*----------------------------------------------------------------------
PLAY
TRAC REND
ADAP SPLI 1 TERM
TRAC REND
GO
ENDPLAY
*----------------------------------------------------------------------
FIN
```

q4gs01.epx

```
Q4GS01
ECHO
```

```
 CONV WIN
LAGR TRID
DIME ADAP NPOI 10 Q4GS 8 ENDA TERM
opti dump dpma
GEOM LIBR POIN 6 Q4GS 2 TERM
 0 0 0  1 0 0  2 0 0
 0 1 0  1 1 0  2 1 0
 1 2 5 4
 2 3 6 5
COMP GROU 1 'stru' LECT 1 PAS 1 2 TERM
     EPAI 1.0 LECT stru _q4gs TERM
     COUL VERT LECT stru TERM
MATE LINE RO 8000. YOUN 2.D11 NU 0.3
          LECT stru _q4gs TERM
LINK COUP ! mandatory for adaptivity
     BLOQ 1 LECT 1 4 TERM
     BLOQ 2 LECT 1 2 3 TERM
ECRI DEPL VITE ACCE FINT FEXT FLIA FDEC CONT ECRO FREQ 1
     FICH ALIC FREQ 1
OPTI NOTE LOG 1
     ADAP DUMP STAT CHEC
     LNKS DUMP STAT DIAG
CALC TINI 0. TEND 1.0 NMAX 4
*-----------------------------------------------------------------------
PLAY
CAME   1 EYE   1.00000E+00  5.00000E-01  5.59017E+00
!         Q     1.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
         VIEW  0.00000E+00  0.00000E+00 -1.00000E+00
         RIGH  1.00000E+00  0.00000E+00  0.00000E+00
         UP    0.00000E+00  1.00000E+00  0.00000E+00
         FOV   2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER :  1.00000E+00  5.00000E-01  0.00000E+00
!RSPHERE:  1.11803E+00
!RADIUS :  5.59017E+00
!ASPECT :  1.00000E+00
!NEAR   :  4.36033E+00
!FAR    :  7.82624E+00
SLER CAM1 1 NFRA 1
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
ADAP SPLI 1 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP SPLI 4 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP USPL 4 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP USPL 1 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ENDPLAY
*-----------------------------------------------------------------------
FIN
```

q4gs02.epx

```
Q4GS02
ECHO
 CONV WIN
LAGR TRID
DIME ADAP NPOI 13 Q4GS 12 ENDA TERM
opti dump dpma
GEOM LIBR POIN 8 Q4GS 3 TERM
 0 0 0  1 0 0  2 0 0
 0 1 0  1 1 0  2 1 0
 1 0 1  1 1 1
 1 2 5 4
 2 3 6 5
 2 5 8 7
COMP GROU 1 'stru' LECT 1 2 3 TERM
     EPAI 1.0 LECT stru _q4gs TERM
     COUL VERT LECT stru TERM
MATE LINE RO 8000. YOUN 2.D11 NU 0.3
          LECT stru _q4gs TERM
LINK COUP ! mandatory for adaptivity
     BLOQ 1 LECT 1 4 TERM
     BLOQ 2 LECT 1 2 3 TERM
ECRI DEPL VITE ACCE FINT FEXT FLIA FDEC CONT ECRO FREQ 1
     FICH ALIC FREQ 1
OPTI NOTE LOG 1
     ADAP DUMP STAT CHEC
     LNKS DUMP STAT DIAG
CALC TINI 0. TEND 1.0 NMAX 6
*-----------------------------------------------------------------------
PLAY
CAME   1 EYE   3.27660E+00 -3.53061E+00  3.49104E+00
!         Q    -8.66182E-01 -4.49496E-01 -1.63098E-01 -1.45195E-01
         VIEW -4.13075E-01  7.31328E-01 -5.42705E-01
         RIGH  9.04635E-01  3.98155E-01 -1.52016E-01
         UP   -1.04907E-01  5.53743E-01  8.26053E-01
         FOV   2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER :  1.00000E+00  5.00000E-01  5.00000E-01
!RSPHERE:  1.22474E+00
!RADIUS :  5.51135E+00
!ASPECT :  1.00000E+00
!NEAR   :  4.28661E+00
!FAR    :  7.96084E+00
SLER CAM1 1 NFRA 1
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
ADAP SPLI 2 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP SPLI 3 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP SPLI 1 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
```

```
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP USPL 1 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP USPL 3 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP USPL 2 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ENDPLAY
*-----------------------------------------------------------------------
FIN
```

q4gs03.dgibi

```
opti echo 1;
opti dime 3 elem cub8;
opti sauv form 'q4gs03.msh';
opti trac psc ftra 'q4gs03_mesh.ps';
p0 = 0 0 0;
p1 = 10 0 0;
p2 = 10 10 0;
p3 = 0 10 0;
c1 = p0 d 10 p1;
c2 = p1 d 10 p2;
c3 = p2 d 10 p3;
c4 = p3 d 10 p0;
mesh = dall c1 c2 c3 c4 plan;
tass mesh;
sauv form mesh;
trac qual mesh;
fin;
```

q4gs03.epx

```
Q4GS03
ECHO
!CONV win
CAST mesh
LAGR TRID
DIME
  ADAP NPOI 3663 Q4GS 4580 ENDA
TERM
GEOM Q4GS mesh TERM
WAVE 2 SPHE X  0 Y  0 T0 0 C 5000 MAXL 4 H1 1.5 H2 5
       SPHE X 10 Y 10 T0 0 C 5000 MAXL 4 H1 1.5 H2 5
COMP EPAI 1.E-3 LECT mesh _q4gs TERM
MATE LINE RO 8000. YOUN 2.E11 NU 0.0
          LECT mesh _q4gs TERM
LINK COUP ! mandatory in adaptivity
ECRI COOR DEPL VITE ACCE FINT FEXT CONT ECRO TFRE 1.E-3
     POIN LECT 1 TERM
     ELEM LECT 1 TERM
     FICH ALIC FREQ 1
OPTI NOTE STEP LIBR
 log 1
CALC TINI 0. TEND 5.0E-3
*=================================================================
PLAY
CAME   1 EYE   5.00000E+00  5.00000E+00  2.82843E+01
!        Q     1.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
         VIEW  0.00000E+00  0.00000E+00 -1.00000E+00
         RIGH  1.00000E+00  0.00000E+00  0.00000E+00
         UP    0.00000E+00  1.00000E+00  0.00000E+00
         FOV   2.48819E+01
scen geom navi free
          face hfro
          line sfre !heou
     colo pape
sler cam1 1 nfra 1
trac offs fich avi nocl nfto 167 fps 25 kfre 10 comp -1 rend
freq 1
gotr loop 165 offs fich avi cont nocl rend
go
trac offs fich avi cont rend
ENDPLAY
*=================================================================
FIN
```

q4gs04.dgibi

```
opti echo 1;
opti dime 3 elem cub8;
opti sauv form 'q4gs04.msh';
opti trac psc ftra 'q4gs04_mesh.ps';
p0 = 0 0 0;
p1 = 10 0 0;
p2 = 10 10 0;
p3 = 0 10 0;
c1 = p0 d 20 p1;
c2 = p1 d 20 p2;
c3 = p2 d 20 p3;
c4 = p3 d 20 p0;
plate = dall c1 c2 c3 c4 plan;
clamp = cont plate;
mesh = plate et clamp;
tass mesh;
sauv form mesh;
trac qual mesh;
trac qual clamp;
fin;
```

q4gs04.epx

```
Q4GS04
ECHO
!CONV win
CAST mesh
LAGR TRID
GEOM Q4GS plate TERM
COMP EPAI 1.E-1 LECT plate TERM
     NGRO 1 'cent' LECT plate TERM COND NEAR POIN 5 5 0
     COUL VERT LECT plate TERM
MATE LINE RO 8000. YOUN 2.E11 NU 0.0
          LECT mesh TERM
LINK COUP SPLT NONE
     BLOQ 123456 LECT clamp TERM
!INIT VITE 3 100.0 LECT plate DIFF clamp TERM
INIT VITE 3 100.0 LECT plate TERM
ECRI COOR DEPL VITE ACCE FINT FEXT CONT ECRO TFRE 1.E-3
     POIN LECT 1 TERM
     ELEM LECT 1 TERM
     FICH ALIC FREQ 1
OPTI NOTE STEP LIBR
 LNKS STAT
 log 1
CALC TINI 0. TEND 50.0E-3
*=================================================================
SUIT
Post-treatment (time curves from alice file)
ECHO
RESU ALIC GARD PSCR
SORT GRAP
AXTE 1.0 'Time [s]'
COUR 1 'dx_cent' DEPL COMP 1 NOEU LECT cent TERM
COUR 2 'dy_cent' DEPL COMP 2 NOEU LECT cent TERM
COUR 3 'dz_cent' DEPL COMP 3 NOEU LECT cent TERM
TRAC 1 2 3 AXES 1.0 'DISPL. [M]' YZER
LIST 1 2 3 AXES 1.0 'DISPL. [M]' YZER
*=================================================================
FIN
```

q4gs05.dgibi

```
opti echo 1;
opti dime 3 elem cub8;
opti sauv form 'q4gs05.msh';
opti trac psc ftra 'q4gs05_mesh.ps';
p0 = 0 0 0;
p1 = 10 0 0;
p2 = 10 10 0;
p3 = 0 10 0;
c1 = p0 d 10 p1;
c2 = p1 d 10 p2;
c3 = p2 d 10 p3;
c4 = p3 d 10 p0;
plate = dall c1 c2 c3 c4 plan;
clamp = cont plate;
mesh = plate et clamp;
tass mesh;
sauv form mesh;
trac qual mesh;
trac qual clamp;
fin;
```

q4gs05.epx

```
Q4GS05
ECHO
!CONV win
CAST mesh
LAGR TRID
DIME ADAP NPOI 320 Q4GS 400 ENDA TERM
GEOM Q4GS plate TERM
COMP EPAI 1.E-1 LECT plate _q4gs TERM
     NGRO 1 'cent' LECT plate TERM COND NEAR POIN 5 5 0
     COUL VERT LECT plate TERM
MATE LINE RO 8000. YOUN 2.E11 NU 0.0
          LECT plate _q4gs TERM
LINK COUP SPLT NONE
     BLOQ 123456 LECT clamp TERM
!INIT VITE 3 100.0 LECT plate DIFF clamp TERM
INIT VITE 3 100.0 LECT plate TERM
     ADAP SPLI LEVE 2 LECT plate TERM
ECRI COOR DEPL VITE ACCE FINT FEXT CONT ECRO TFRE 1.E-3
     POIN LECT 1 TERM
     ELEM LECT 1 TERM
     FICH ALIC FREQ 1
OPTI NOTE STEP LIBR
 LNKS STAT
 log 1
CALC TINI 0. TEND 50.0E-3
*=================================================================
SUIT
Post-treatment (time curves from alice file)
ECHO
RESU ALIC GARD PSCR
SORT GRAP
AXTE 1.0 'Time [s]'
COUR 1 'dx_cent' DEPL COMP 1 NOEU LECT cent TERM
COUR 2 'dy_cent' DEPL COMP 2 NOEU LECT cent TERM
COUR 3 'dz_cent' DEPL COMP 3 NOEU LECT cent TERM
TRAC 1 2 3 AXES 1.0 'DISPL. [M]' YZER
LIST 1 2 3 AXES 1.0 'DISPL. [M]' YZER
RCOU 11 'dx_cent' FICH 'q4gs04.pun' RENA 'dx_cent_04'
RCOU 12 'dy_cent' FICH 'q4gs04.pun' RENA 'dy_cent_04'
RCOU 13 'dz_cent' FICH 'q4gs04.pun' RENA 'dz_cent_04'
TRAC 3 13 AXES 1.0 'DISPL. [M]' YZER
COLO NOIR ROUG
*=================================================================
FIN
```

q4gs06.dgibi

```
opti echo 1;
opti dime 3 elem cub8;
opti sauv form 'q4gs06.msh';
opti trac psc ftra 'q4gs06_mesh.ps';
p0 = 0 0 0;
p1 = 10 0 0;
p2 = 10 10 0;
p3 = 0 10 0;
c1 = p0 d 10 p1;
c2 = p1 d 10 p2;
c3 = p2 d 10 p3;
c4 = p3 d 10 p0;
plate = dall c1 c2 c3 c4 plan;
clamp = cont plate;
mesh = plate et clamp;
tass mesh;
sauv form mesh;
trac qual mesh;
trac qual clamp;
fin;
```

q4gs06.epx

```
Q4GS06
ECHO
!CONV win
CAST mesh
LAGR TRID
DIME ADAP NPOI 320 Q4GS 400 ENDA TERM
GEOM Q4GS plate TERM
WAVE 2 SPHE X  0 Y  0 T0 0 C 5000 MAXL 2 H1 1.5 H2 5
       SPHE X 10 Y 10 T0 0 C 5000 MAXL 2 H1 1.5 H2 5
COMP EPAI 1.E-1 LECT plate _q4gs TERM
     NGRO 1 'cent' LECT plate TERM COND NEAR POIN 5 5 0
     COUL VERT LECT plate TERM
MATE LINE RO 8000. YOUN 2.E11 NU 0.0
          LECT plate _q4gs TERM
LINK COUP SPLT NONE
     BLOQ 123456 LECT clamp TERM
!INIT VITE 3 100.0 LECT plate DIFF clamp TERM
INIT VITE 3 100.0 LECT plate TERM
ECRI COOR DEPL VITE ACCE FINT FEXT CONT ECRO TFRE 1.E-3
     POIN LECT 1 TERM
     ELEM LECT 1 TERM
     FICH ALIC FREQ 1
OPTI NOTE STEP LIBR
 LNKS STAT
 log 1
CALC TINI 0. TEND 50.0E-3
*=================================================================
SUIT
Post-treatment (time curves from alice file)
ECHO
RESU ALIC GARD PSCR
SORT GRAP
AXTE 1.0 'Time [s]'
COUR 1 'dx_cent' DEPL COMP 1 NOEU LECT cent TERM
COUR 2 'dy_cent' DEPL COMP 2 NOEU LECT cent TERM
COUR 3 'dz_cent' DEPL COMP 3 NOEU LECT cent TERM
TRAC 1 2 3 AXES 1.0 'DISPL. [M]' YZER
LIST 1 2 3 AXES 1.0 'DISPL. [M]' YZER
RCOU 11 'dx_cent' FICH 'q4gs04.pun' RENA 'dx_cent_04'
RCOU 12 'dy_cent' FICH 'q4gs04.pun' RENA 'dy_cent_04'
RCOU 13 'dz_cent' FICH 'q4gs04.pun' RENA 'dz_cent_04'
TRAC 3 13 AXES 1.0 'DISPL. [M]' YZER
COLO NOIR ROUG
*=================================================================
FIN
```

q4gs07.epx

```
Q4GS07
ECHO
 CONV WIN
LAGR TRID
DIME ADAP NPOI 10 Q4GS 8 CL3D 8 ENDA TERM
opti dump dpma
GEOM LIBR POIN 4 Q4GS 1 CL3D 1 TERM
 0 0 0  1 0 0  1 1 0  0 1 0
 1 2 3 4
 1 4 3 2
COMP GROU 2 'stru' LECT 1 TERM
            'clxx' LECT 2 TERM
     EPAI 1.0 LECT stru _q4gs TERM
     COUL VERT LECT stru TERM
          JAUN LECT clxx TERM
MATE LINE RO 8000. YOUN 2.D11 NU 0.3
          LECT stru _q4gs TERM
     IMPE PIMP RO 8000 PRES 1.E5 PREF 0
          TABP 2 0 1 100 1
          LECT clxx _cl3d TERM
LINK COUP ! mandatory for adaptivity
ECRI DEPL VITE ACCE FINT FEXT FLIA FDEC CONT ECRO FREQ 1
     FICH ALIC FREQ 1
OPTI NOTE LOG 1
     ADAP DUMP STAT CHEC
     LNKS DUMP STAT DIAG
CALC TINI 0. TEND 1.0 NMAX 4
*-----------------------------------------------------------------
PLAY
CAME   1 EYE   5.00000E-01  5.00000E-01  3.53553E+00
!        Q     1.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
         VIEW  0.00000E+00  0.00000E+00 -1.00000E+00
         RIGH  1.00000E+00  0.00000E+00  0.00000E+00
         UP    0.00000E+00  1.00000E+00  0.00000E+00
         FOV   2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER :  5.00000E-01  5.00000E-01  0.00000E+00
!RSPHERE:  7.07107E-01
!RADIUS :  3.53553E+00
!ASPECT :  1.00000E+00
!NEAR   :  2.75772E+00
!FAR    :  4.94975E+00
SLER CAM1 1 NFRA 1
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          LINE SIOU
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          LINE SIOU
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
ADAP SPLI 1 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          LINE SIOU
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          LINE SIOU
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP SPLI 4 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          LINE SIOU
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          LINE SIOU
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP USPL 4 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          LINE SIOU
          POIN SPHE 1
     TEXT NODE ELEM
```

```
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          LINE SIOU
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP USPL 1 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          LINE SIOU
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          LINE SIOU
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ENDPLAY
*----------------------------------------------------------------------
FIN
```

q4gs08.epx

```
Q4GS08
ECHO
 CONV WIN
LAGR TRID
DIME ADAP NPOI 10 Q4GS 8 CL3D 8 ENDA TERM
opti dump dpma
GEOM LIBR POIN 4 Q4GS 1 CL3D 1 TERM
 0 0 0  1 0 0  1 1 0  0 1 0
 1 2 3 4
 1 2 3 4
COMP GROU 2 'stru' LECT 1 TERM
            'clxx' LECT 2 TERM
     EPAI 1.0 LECT stru _q4gs TERM
     COUL VERT LECT stru TERM
          JAUN LECT clxx TERM
MATE LINE RO 8000. YOUN 2.D11 NU 0.3
          LECT stru _q4gs TERM
     IMPE PIMP RO 8000 PRES 1.E5 PREF 0
          TABP 2 0 1 100 1
          LECT clxx _cl3d TERM
LINK COUP ! mandatory for adaptivity
ECRI DEPL VITE ACCE FINT FEXT FLIA FDEC CONT ECRO FREQ 1
     FICH ALIC FREQ 1
OPTI NOTE LOG 1
     ADAP DUMP STAT CHEC
     LNKS DUMP STAT DIAG
CALC TINI 0. TEND 1.0 NMAX 4
*----------------------------------------------------------------------
PLAY
CAME   1 EYE   5.00000E-01  5.00000E-01  3.53553E+00
!        Q     1.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
         VIEW  0.00000E+00  0.00000E+00 -1.00000E+00
         RIGH  1.00000E+00  0.00000E+00  0.00000E+00
         UP    0.00000E+00  1.00000E+00  0.00000E+00
         FOV   2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER :  5.00000E-01  5.00000E-01  0.00000E+00
!RSPHERE:  7.07107E-01
!RADIUS :  3.53553E+00
!ASPECT :  1.00000E+00
!NEAR   :  2.75772E+00
!FAR    :  4.94975E+00
SLER CAM1 1 NFRA 1
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          LINE SIOU
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          LINE SIOU
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
ADAP SPLI 1 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          LINE SIOU
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          LINE SIOU
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP SPLI 4 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          LINE SIOU
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          LINE SIOU
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP USPL 4 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          LINE SIOU
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          LINE SIOU
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP USPL 1 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          LINE SIOU
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          LINE SIOU
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ENDPLAY
*----------------------------------------------------------------------
FIN
```

q4gs09.dgibi

```
opti echo 1;
opti dime 3 elem cub8;
opti sauv form 'q4gs09.msh';
opti trac psc ftra 'q4gs09_mesh.ps';
p0 = 0 0 0;
p1 = 10 0 0;
p2 = 10 10 0;
p3 = 0 10 0;
c1 = p0 d 20 p1;
c2 = p1 d 20 p2;
c3 = p2 d 20 p3;
c4 = p3 d 20 p0;
tol = 0.001;
plate = dall c1 c2 c3 c4 plan;
clamp = cont plate;
pres = plate plus (0 0 0);
elim tol (plate et pres);
mesh = plate et pres et clamp;
tass mesh;
sauv form mesh;
trac qual mesh;
trac qual clamp;
fin;
```

q4gs09.epx

```
Q4GS09
ECHO
!CONV win
CAST mesh
LAGR TRID
GEOM Q4GS plate CL3D pres TERM
COMP EPAI 1.E-1 LECT plate TERM
     NGRO 1 'cent' LECT plate TERM COND NEAR POIN 5 5 0
     COUL VERT LECT plate TERM
          JAUN LECT pres TERM
MATE LINE RO 8000. YOUN 2.E11 NU 0.0
          LECT plate TERM
     IMPE PIMP RO 8000 PRES 10.E5 PREF 0
          TABP 2 0 1 100 1
          LECT pres TERM
LINK COUP SPLT NONE
     BLOQ 123456 LECT clamp TERM
```

```
ECRI COOR DEPL VITE ACCE FINT FEXT CONT ECRO TFRE 1.E-3
     POIN LECT 1 TERM
     ELEM LECT 1 TERM
     FICH ALIC FREQ 1
OPTI NOTE STEP LIBR
 LNKS STAT
 log 1
CALC TINI 0. TEND 60.0E-3
*================================================================
SUIT
Post-treatment (time curves from alice file)
ECHO
RESU ALIC GARD PSCR
SORT GRAP
AXTE 1.0 'Time [s]'
COUR 1 'dx_cent' DEPL COMP 1 NOEU LECT cent TERM
COUR 2 'dy_cent' DEPL COMP 2 NOEU LECT cent TERM
COUR 3 'dz_cent' DEPL COMP 3 NOEU LECT cent TERM
TRAC 1 2 3 AXES 1.0 'DISPL. [M]' YZER
LIST 1 2 3 AXES 1.0 'DISPL. [M]' YZER
*================================================================
FIN
```

q4gs10.dgibi

```
opti echo 1;
opti dime 3 elem cub8;
opti sauv form 'q4gs10.msh';
opti trac psc ftra 'q4gs10_mesh.ps';
p0 = 0 0 0;
p1 = 10 0 0;
p2 = 10 10 0;
p3 = 0 10 0;
c1 = p0 d 10 p1;
c2 = p1 d 10 p2;
c3 = p2 d 10 p3;
c4 = p3 d 10 p0;
tol = 0.001;
plate = dall c1 c2 c3 c4 plan;
clamp = cont plate;
pres = plate plus (0 0 0);
elim tol (plate et pres);
mesh = plate et pres et clamp;
tass mesh;
sauv form mesh;
trac qual mesh;
trac qual clamp;
fin;
```

q4gs10.epx

```
Q4GS10
ECHO
!CONV win
CAST mesh
LAGR TRID
DIME ADAP NPOI 320 Q4GS 400 CL3D 400 ENDA TERM
GEOM Q4GS plate CL3D pres TERM
COMP EPAI 1.E-1 LECT plate _q4gs TERM
     NGRO 1 'cent' LECT plate TERM COND NEAR POIN 5 5 0
     COUL VERT LECT plate TERM
          JAUN LECT pres TERM
MATE LINE RO 8000. YOUN 2.E11 NU 0.0
          LECT plate _q4gs TERM
     IMPE PIMP RO 8000 PRES 10.E5 PREF 0
          TABP 2 0 1 100 1
          LECT pres _cl3d TERM
LINK COUP SPLT NONE
     BLOQ 123456 LECT clamp TERM
INIT ADAP SPLI LEVE 2 LECT plate TERM
ECRI COOR DEPL VITE ACCE FINT FEXT CONT ECRO TFRE 1.E-3
     POIN LECT 1 TERM
     ELEM LECT 1 TERM
     FICH ALIC FREQ 1
OPTI NOTE STEP LIBR
 LNKS STAT
 log 1
CALC TINI 0. TEND 60.0E-3
*================================================================
SUIT
Post-treatment (time curves from alice file)
ECHO
RESU ALIC GARD PSCR
SORT GRAP
AXTE 1.0 'Time [s]'
COUR 1 'dx_cent' DEPL COMP 1 NOEU LECT cent TERM
COUR 2 'dy_cent' DEPL COMP 2 NOEU LECT cent TERM
COUR 3 'dz_cent' DEPL COMP 3 NOEU LECT cent TERM
TRAC 1 2 3 AXES 1.0 'DISPL. [M]' YZER
LIST 1 2 3 AXES 1.0 'DISPL. [M]' YZER
RCOU 11 'dx_cent' FICH 'q4gs09.pun' RENA 'dx_cent_09'
RCOU 12 'dy_cent' FICH 'q4gs09.pun' RENA 'dy_cent_09'
RCOU 13 'dz_cent' FICH 'q4gs09.pun' RENA 'dz_cent_09'
TRAC 3 13 AXES 1.0 'DISPL. [M]' YZER
COLO NOIR ROUG
*================================================================
FIN
```

t3gs01.epx

```
T3GS01
ECHO
 CONV WIN
LAGR TRID
DIME ADAP NPOI 6 T3GS 8 ENDA TERM
opti dump dpma
GEOM LIBR POIN 4 T3GS 2 TERM
 0 0 0  1 0 0  2 0 0
 1 1 0
 1 2 4
 2 3 4
COMP GROU 1 'stru' LECT 1 PAS 1 2 TERM
     EPAI 1.0 LECT stru _t3gs TERM
     COUL VERT LECT stru TERM
MATE LINE RO 8000. YOUN 2.D11 NU 0.3
          LECT stru _t3gs TERM
LINK COUP ! mandatory for adaptivity
     BLOQ 1 LECT 2 4 TERM
     BLOQ 2 LECT 1 2 3 TERM
ECRI DEPL VITE ACCE FINT FEXT FLIA FDEC CONT ECRO FREQ 1
     FICH ALIC FREQ 1
OPTI NOTE LOG 1
     ADAP DUMP STAT CHEC
     LNKS DUMP STAT DIAG
CALC TINI 0. TEND 1.0 NMAX 4
*-------------------------------------------------------------------
PLAY
CAME   1 EYE   1.00000E+00  5.00000E-01  5.59017E+00
!        Q     1.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
         VIEW  0.00000E+00  0.00000E+00 -1.00000E+00
         RIGH  1.00000E+00  0.00000E+00  0.00000E+00
         UP    0.00000E+00  1.00000E+00  0.00000E+00
         FOV   2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER :  1.00000E+00  5.00000E-01  0.00000E+00
!RSPHERE:  1.11803E+00
!RADIUS :  5.59017E+00
!ASPECT :  1.00000E+00
!NEAR   :  4.36033E+00
!FAR    :  7.82624E+01
SLER CAM1 1 NFRA 1
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
ADAP SPLI 1 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP SPLI 2 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP USPL 2 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP USPL 1 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
```

```
ENDPLAY
*----------------------------------------------------------------------
FIN
```

t3gs02.epx

```
T3GS02
ECHO
 CONV WIN
LAGR TRID
DIME ADAP NPOI 9 T3GS 12 ENDA TERM
opti dump dpma
GEOM LIBR POIN 5 T3GS 3 TERM
 0 0 0  1 0 0  2 0 0
 1 1 0
 1 1 1
 1 2 4
 2 3 4
 2 4 5
COMP GROU 1 'stru' LECT 1 2 3 TERM
     EPAI 1.0 LECT stru _t3gs TERM
     COUL VERT LECT stru TERM
MATE LINE RO 8000. YOUN 2.D11 NU 0.3
          LECT stru _t3gs TERM
LINK COUP ! mandatory for adaptivity
     BLOQ 1 LECT 2 4 TERM
     BLOQ 2 LECT 1 2 3 TERM
ECRI DEPL VITE ACCE FINT FEXT FLIA FDEC CONT ECRO FREQ 1
     FICH ALIC FREQ 1
OPTI NOTE LOG 1
     ADAP DUMP STAT CHEC
     LNKS DUMP STAT DIAG
CALC TINI 0. TEND 1.0 NMAX 6
*----------------------------------------------------------------------
PLAY
CAME   1 EYE   3.27660E+00 -3.53061E+00  3.49104E+00
!        Q    -8.66182E-01 -4.49496E-01 -1.63098E-01 -1.45195E-01
         VIEW -4.13075E-01  7.31328E-01 -5.42705E-01
         RIGH  9.04635E-01  3.98155E-01 -1.52016E-01
         UP   -1.04907E-01  5.53743E-01  8.26053E-01
         FOV   2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER :  1.00000E+00  5.00000E-01  5.00000E-01
!RSPHERE:  1.22474E+00
!RADIUS :  5.51135E+00
!ASPECT :  1.00000E+00
!NEAR   :  4.28661E+00
!FAR    :  7.96084E+00
SLER CAM1 1 NFRA 1
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
ADAP SPLI 2 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP SPLI 3 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP SPLI 1 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP USPL 1 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP USPL 3 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ADAP USPL 2 TERM
SCEN OBJE SELV HANG
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
SCEN OBJE SELV BHAN
     GEOM NAVI FREE
          FACE HFRO
          POIN SPHE 1
     TEXT NODE ELEM
     COLO PAPE
TRAC REND
TRAC OFFS FICH BMP REND
GO
ENDPLAY
*----------------------------------------------------------------------
FIN
```

t3gs03.dgibi

```
opti echo 1;
opti dime 3 elem cub8;
opti sauv form 't3gs03.msh';
opti trac psc ftra 't3gs03_mesh.ps';
p0 = 0 0 0;
p1 = 10 0 0;
p2 = 10 10 0;
p3 = 0 10 0;
c1 = p0 d 10 p1;
c2 = p1 d 10 p2;
c3 = p2 d 10 p3;
c4 = p3 d 10 p0;
mesh4 = dall c1 c2 c3 c4 plan;
mesh = chan tri3 mesh4;
tass mesh;
sauv form mesh;
oeil = 0 0 100000;
trac oeil qual mesh;
fin;
```

t3gs03.epx

```
T3GS03
ECHO
!CONV win
CAST mesh
LAGR TRID
DIME
  ADAP NPOI 3609 T3GS 9068 ENDA
TERM
GEOM T3GS mesh TERM
WAVE 2 SPHE X  0 Y  0 T0 0 C 5000 MAXL 4 H1 1.5 H2 5
       SPHE X 10 Y 10 T0 0 C 5000 MAXL 4 H1 1.5 H2 5
COMP EPAI 1.E-3 LECT mesh _t3gs TERM
MATE LINE RO 8000. YOUN 2.E11 NU 0.0
          LECT mesh _t3gs TERM
LINK COUP ! mandatory in adaptivity
ECRI COOR DEPL VITE ACCE FINT FEXT CONT ECRO TFRE 1.E-3
     POIN LECT 1 TERM
     ELEM LECT 1 TERM
     FICH ALIC FREQ 1
OPTI NOTE STEP LIBR
 log 1
CALC TINI 0. TEND 5.0E-3
*=================================================================
PLAY
CAME   1 EYE   5.00000E+00  5.00000E+00  2.82843E+01
!        Q     1.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
         VIEW  0.00000E+00  0.00000E+00 -1.00000E+00
         RIGH  1.00000E+00  0.00000E+00  0.00000E+00
```

```
         UP    0.00000E+00  1.00000E+00  0.00000E+00
         FOV   2.48819E+01
scen geom navi free
          face hfro
          line sfre !heou
     colo pape
sler cam1 1 nfra 1
trac offs fich avi nocl nfto 236 fps 25 kfre 10 comp -1 rend
freq 1
gotr loop 234 offs fich avi cont nocl rend
go
trac offs fich avi cont rend
ENDPLAY
*=================================================================
FIN
```

t3gs09.dgibi

```
opti echo 1;
opti dime 3 elem cub8;
opti sauv form 't3gs09.msh';
opti trac psc ftra 't3gs09_mesh.ps';
p0 = 0 0 0;
p1 = 10 0 0;
p2 = 10 10 0;
p3 = 0 10 0;
c1 = p0 d 20 p1;
c2 = p1 d 20 p2;
c3 = p2 d 20 p3;
c4 = p3 d 20 p0;
tol = 0.001;
plate4 = dall c1 c2 c3 c4 plan;
plate = chan tri3 plate4;
clamp = cont plate;
pres = plate plus (0 0 0);
elim tol (plate et pres);
mesh = plate et pres et clamp;
tass mesh;
sauv form mesh;
oeil = 0 0 100000;
trac oeil qual mesh;
trac oeil qual clamp;
fin;
```

t3gs09.epx

```
T3GS09
ECHO
!CONV win
CAST mesh
LAGR TRID
GEOM T3GS plate CL3T pres TERM
COMP EPAI 1.E-1 LECT plate TERM
     NGRO 1 'cent' LECT plate TERM COND NEAR POIN 5 5 0
     COUL VERT LECT plate TERM
          JAUN LECT pres TERM
MATE LINE RO 8000. YOUN 2.E11 NU 0.0
          LECT plate TERM
     IMPE PIMP RO 8000 PRES 10.E5 PREF 0
          TABP 2 0 1 100 1
          LECT pres TERM
LINK COUP SPLT NONE
     BLOQ 123456 LECT clamp TERM
ECRI COOR DEPL VITE ACCE FINT FEXT CONT ECRO TFRE 1.E-3
     POIN LECT 1 TERM
     ELEM LECT 1 TERM
     FICH ALIC FREQ 1
OPTI NOTE STEP LIBR
 LNKS STAT
 log 1
CALC TINI 0. TEND 60.0E-3
*=================================================================
SUIT
Post-treatment (time curves from alice file)
ECHO
RESU ALIC GARD PSCR
SORT GRAP
AXTE 1.0 'Time [s]'
COUR 1 'dx_cent' DEPL COMP 1 NOEU LECT cent TERM
COUR 2 'dy_cent' DEPL COMP 2 NOEU LECT cent TERM
COUR 3 'dz_cent' DEPL COMP 3 NOEU LECT cent TERM
TRAC 1 2 3 AXES 1.0 'DISPL. [M]' YZER
LIST 1 2 3 AXES 1.0 'DISPL. [M]' YZER
*=================================================================
FIN
```

t3gs10.dgibi

```
opti echo 1;
opti dime 3 elem cub8;
opti sauv form 't3gs10.msh';
opti trac psc ftra 't3gs10_mesh.ps';
p0 = 0 0 0;
p1 = 10 0 0;
p2 = 10 10 0;
p3 = 0 10 0;
c1 = p0 d 10 p1;
c2 = p1 d 10 p2;
c3 = p2 d 10 p3;
c4 = p3 d 10 p0;
tol = 0.001;
plate4 = dall c1 c2 c3 c4 plan;
plate = chan tri3 plate4;
clamp = cont plate;
pres = plate plus (0 0 0);
elim tol (plate et pres);
mesh = plate et pres et clamp;
tass mesh;
sauv form mesh;
oeil = 0 0 100000;
trac oeil qual mesh;
trac oeil qual clamp;
fin;
```

t3gs10.epx

```
T3GS10
ECHO
!CONV win
CAST mesh
LAGR TRID
DIME ADAP NPOI 320 T3GS 800 CL3T 800 ENDA TERM
GEOM T3GS plate CL3T pres TERM
COMP EPAI 1.E-1 LECT plate _t3gs TERM
     NGRO 1 'cent' LECT plate TERM COND NEAR POIN 5 5 0
     COUL VERT LECT plate TERM
          JAUN LECT pres TERM
MATE LINE RO 8000. YOUN 2.E11 NU 0.0
          LECT plate _t3gs TERM
     IMPE PIMP RO 8000 PRES 10.E5 PREF 0
          TABP 2 0 1 100 1
          LECT pres _cl3t TERM
LINK COUP SPLT NONE
     BLOQ 123456 LECT clamp TERM
INIT ADAP SPLI LEVE 2 LECT plate TERM
ECRI COOR DEPL VITE ACCE FINT FEXT CONT ECRO TFRE 1.E-3
     POIN LECT 1 TERM
     ELEM LECT 1 TERM
     FICH ALIC FREQ 1
OPTI NOTE STEP LIBR
 LNKS STAT
 log 1
CALC TINI 0. TEND 60.0E-3
*=================================================================
SUIT
Post-treatment (time curves from alice file)
ECHO
RESU ALIC GARD PSCR
SORT GRAP
AXTE 1.0 'Time [s]'
COUR 1 'dx_cent' DEPL COMP 1 NOEU LECT cent TERM
COUR 2 'dy_cent' DEPL COMP 2 NOEU LECT cent TERM
COUR 3 'dz_cent' DEPL COMP 3 NOEU LECT cent TERM
TRAC 1 2 3 AXES 1.0 'DISPL. [M]' YZER
LIST 1 2 3 AXES 1.0 'DISPL. [M]' YZER
RCOU 11 'dx_cent' FICH 't3gs09.pun' RENA 'dx_cent_09'
RCOU 12 'dy_cent' FICH 't3gs09.pun' RENA 'dy_cent_09'
RCOU 13 'dz_cent' FICH 't3gs09.pun' RENA 'dz_cent_09'
TRAC 3 13 AXES 1.0 'DISPL. [M]' YZER
COLO NOIR ROUG
*=================================================================
FIN
```

ve1d01.epx

```
VE1D01
ECHO
!CONV win
LAGR CPLA
GEOM 9 '(6E12.5)' '(12I6)' POIN 1449 Q41L 1280 TERM
COMP GROU 1 'mesh' LECT 1 PAS 1 1280 TERM
     COUL VERT LECT mesh TERM
     EPAI 1. LECT mesh TERM
MATE VM23 RO 8000. YOUN 2.E11 NU 0.0 ELAS 2.E11
          TRAC 1 2.E11 1.D0
          LECT mesh TERM
INIT VITE FILE 've1d01.vit'
ECRI DEPL VITE TFRE 40.E-6
     FICH ALIC FREQ 1
OPTI NOTE LOG 1
CALC TINI 0. TEND 40.0E-6
FIN
```

ve1d01b.epx

```
VE1D01B
ECHO
RESU ALIC 've1d01.ali' GARD PSCR
SORT VISU NSTO 1
*=======================================================================
PLAY
CAME   1 EYE   4.00000E-01  2.00000E-02  2.00250E+00
!        Q     1.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
         VIEW  0.00000E+00  0.00000E+00 -1.00000E+00
         RIGH  1.00000E+00  0.00000E+00  0.00000E+00
         UP    0.00000E+00  1.00000E+00  0.00000E+00
         FOV   2.48819E+01
SCEN GEOM NAVI FREE
!    FACE HFRO
     LINE HEOU SFRE
!    VECT SCCO FIEL VITE SCAL USER PROG 0.07 PAS 0.07 0.98
!    TEXT VSCA
     ISO  FILL FIEL VITE 1 SCAL USER PROG -0.975 PAS 0.15 0.975
!    ISO  FILL FIEL VITE 2 SCAL USER PROG -0.975 PAS 0.15 0.975
     TEXT ISCA
     COLO PAPE
SLER CAM1 1 NFRA 1
TRAC OFFS FICH AVI NOCL NFTO 51 FPS 15 KFRE 10 COMP -1 REND
FREQ 1
GOTR LOOP 49 OFFS FICH AVI CONT NOCL REND
GO
TRAC OFFS FICH AVI CONT REND
ENDPLAY
*=======================================================================
FIN
```

ve1d01c.epx

```
VE1D01C
ECHO
```

```
RESU ALIC 've1d01.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT 1 PAS 1 161 TERM
SORT GRAP
PERF 've1d01c.pun'
AXTE 1.0 'Time [s]'
SCOU  1 'v_01'  T 0.00E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          VITE COMP 1
SCOU  2 'v_02'  T 0.01E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          VITE COMP 1
SCOU  3 'v_03'  T 0.02E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          VITE COMP 1
SCOU  4 'v_04'  T 0.03E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          VITE COMP 1
SCOU  5 'v_05'  T 0.04E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          VITE COMP 1
TRAC  1  2  3  4  5 AXES 1.0 'VELOC. [M/S]'
LIST  1  2  3  4  5 AXES 1.0 'VELOC. [M/S]'
TRAC  1 5 AXES 1.0 'VELOC. [M/S]'
COLO NOIR ROUG
SCOU 11 'd_01'  T 0.00E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          DEPL COMP 1
SCOU 12 'd_02'  T 0.01E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          DEPL COMP 1
SCOU 13 'd_03'  T 0.02E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          DEPL COMP 1
SCOU 14 'd_04'  T 0.03E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          DEPL COMP 1
SCOU 15 'd_05'  T 0.04E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          DEPL COMP 1
TRAC 11 12 13 14 15 AXES 1.0 'DISPL. [M]'
LIST 11 12 13 14 15 AXES 1.0 'DISPL. [M]'
TRAC 11 15 AXES 1.0 'VELOC. [M/S]'
COLO NOIR ROUG
*=================================================================
FIN
```

ve1d17.epx

```
VE1D17
ECHO
!CONV win
LAGR CPLA
DIME
  ADAP NPOI 104 FUN2 208 ENDA
TERM
GEOM LIBR POIN 21 FUN2 20 TERM
 0.00000E+00 0.00000E+00 4.00000E-02 0.00000E+00 8.00000E-02 0.00000E+00
 1.20000E-01 0.00000E+00 1.60000E-01 0.00000E+00 2.00000E-01 0.00000E+00
 2.40000E-01 0.00000E+00 2.80000E-01 0.00000E+00 3.20000E-01 0.00000E+00
 3.60000E-01 0.00000E+00 4.00000E-01 0.00000E+00 4.40000E-01 0.00000E+00
 4.80000E-01 0.00000E+00 5.20000E-01 0.00000E+00 5.60000E-01 0.00000E+00
 6.00000E-01 0.00000E+00 6.40000E-01 0.00000E+00 6.80000E-01 0.00000E+00
 7.20000E-01 0.00000E+00 7.60000E-01 0.00000E+00 8.00000E-01 0.00000E+00
     1     2     2     3     3     4
     4     5     5     6     6     7
     7     8     8     9     9    10
    10    11    11    12    12    13
    13    14    14    15    15    16
    16    17    17    18    18    19
    19    20    20    21
COMP GROU 2 'mesh' LECT 1 PAS 1 20 TERM
            'coro' LECT mesh TERM COND BOX X0 0.3  Y0 -0.02
                                       DX 0.16 DY  0.02
     COUL VERT LECT mesh _fun2 TERM
          ROUG LECT coro TERM
     EPAI 1. LECT mesh _fun2 TERM
ADAP INDI VITE
     PELE 1000 ALFA 4
     CERR 1.0 LECT mesh TERM
MATE VM23 RO 8000. YOUN 2.E11 NU 0.0 ELAS 2.E11
          TRAC 1 2.E11 1.D0
          LECT mesh _fun2 TE0RM
LINK COUP ! necessaire en adaptivite
INIT ADAP SPLI LEVE 4 LECT coro TERM
     ROUT CASE 3
ECRI COOR DEPL VITE FREQ 1
     FICH ALIC FREQ 1
OPTI NOTE LOG 1
  ADAP STAT RCON
       MAXL 4 DUMP
  LNKS STAT
CALC TINI 0. TEND 40.0E-6
FIN
```

ve1d17a.epx

```
VE1D17A
ECHO
RESU ALIC 've1d17.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT 1 PAS 1 21 TERM
SORT GRAP
PERF 've1d17a.pun'
AXTE 1.0 'Time [s]'
SCOU  1 'v_01'  T 0.00E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                          VITE COMP 1
SCOU  2 'v_02'  T 0.01E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                          VITE COMP 1
SCOU  3 'v_03'  T 0.02E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                          VITE COMP 1
SCOU  4 'v_04'  T 0.03E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                          VITE COMP 1
SCOU  5 'v_05'  T 0.04E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                          VITE COMP 1
TRAC  1  2  3  4  5 AXES 1.0 'VELOC. [M/S]'
LIST  1  2  3  4  5 AXES 1.0 'VELOC. [M/S]'
*=================================================================
FIN
```

ve1d17c.epx

```
VE1D17C
ECHO
RESU ALIC 've1d17.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT 1 PAS 1 21 TERM
SORT GRAP
PERF 've1d17c.pun'
AXTE 1.0 'Time [s]'
SCOU  0 'v_0' NPAS 0 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  1 'v_1' NPAS 1 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  2 'v_2' NPAS 2 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  3 'v_3' NPAS 3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  4 'v_4' NPAS 4 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU 10 'l_0' NPAS 0 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 11 'l_1' NPAS 1 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 12 'l_2' NPAS 2 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 13 'l_3' NPAS 3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 14 'l_4' NPAS 4 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
COUR 20 '10-1' SUBC 10 1.0
COUR 21 '11-1' SUBC 11 1.0
COUR 22 '12-1' SUBC 12 1.0
COUR 23 '13-1' SUBC 13 1.0
COUR 24 '14-1' SUBC 14 1.0
COUR 30 '(10-1)/3' DIVC 20 3.0
COUR 31 '(11-1)/3' DIVC 21 3.0
COUR 32 '(12-1)/3' DIVC 22 3.0
COUR 33 '(13-1)/3' DIVC 23 3.0
COUR 34 '(14-1)/3' DIVC 24 3.0
TRAC  0  1  2  3  4 AXES 1.0 'VELOC. [M/S]'
COLO NOIR TURQ VERT ROUG ROSE
TRAC 0 30 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 1 31 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 2 32 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 3 33 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 4 34 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
SCOU 100 'cerr_0' NPAS 0 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         CERR
SCOU 101 'cerr_1' NPAS 1 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         CERR
SCOU 102 'cerr_2' NPAS 2 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         CERR
SCOU 103 'cerr_3' NPAS 3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         CERR
SCOU 104 'cerr_4' NPAS 4 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         CERR
TRAC 100 AXES 1.0 'CERR'
TRAC 101 AXES 1.0 'CERR'
TRAC 102 AXES 1.0 'CERR'
TRAC 103 AXES 1.0 'CERR'
TRAC 104 AXES 1.0 'CERR'
SCOU 110 'maxc_0' NPAS 0 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         MAXC
SCOU 111 'maxc_1' NPAS 1 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         MAXC
SCOU 112 'maxc_2' NPAS 2 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         MAXC
SCOU 113 'maxc_3' NPAS 3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         MAXC
SCOU 114 'maxc_4' NPAS 4 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         MAXC
TRAC 110 AXES 1.0 'MAXC'
TRAC 111 AXES 1.0 'MAXC'
TRAC 112 AXES 1.0 'MAXC'
TRAC 113 AXES 1.0 'MAXC'
TRAC 114 AXES 1.0 'MAXC'
SCOU 120 'erri_0' NPAS 0 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         ERRI
SCOU 121 'erri_1' NPAS 1 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         ERRI
SCOU 122 'erri_2' NPAS 2 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         ERRI
SCOU 123 'erri_3' NPAS 3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         ERRI
SCOU 124 'erri_4' NPAS 4 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         ERRI
TRAC 120 AXES 1.0 'ERRI'
TRAC 121 AXES 1.0 'ERRI'
TRAC 122 AXES 1.0 'ERRI'
TRAC 123 AXES 1.0 'ERRI'
TRAC 124 AXES 1.0 'ERRI'
SCOU 130 'clen_0' NPAS 0 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         CLEN
SCOU 131 'clen_1' NPAS 1 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         CLEN
SCOU 132 'clen_2' NPAS 2 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         CLEN
SCOU 133 'clen_3' NPAS 3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         CLEN
SCOU 134 'clen_4' NPAS 4 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         CLEN
TRAC 130 AXES 1.0 'CLEN'
TRAC 131 AXES 1.0 'CLEN'
TRAC 132 AXES 1.0 'CLEN'
TRAC 133 AXES 1.0 'CLEN'
TRAC 134 AXES 1.0 'CLEN'
SCOU 140 'ilen_0' NPAS 0 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         ILEN
SCOU 141 'ilen_1' NPAS 1 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         ILEN
SCOU 142 'ilen_2' NPAS 2 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         ILEN
SCOU 143 'ilen_3' NPAS 3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         ILEN
SCOU 144 'ilen_4' NPAS 4 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         ILEN
TRAC 140 AXES 1.0 'ILEN'
TRAC 141 AXES 1.0 'ILEN'
TRAC 142 AXES 1.0 'ILEN'
TRAC 143 AXES 1.0 'ILEN'
TRAC 144 AXES 1.0 'ILEN'
```

```
*================================================================
FIN
```

ve1d17d.epx

```
VE1D17C
ECHO
RESU ALIC 've1d17.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT 1 PAS 1 21 TERM
SORT GRAP
PERF 've1d17c.pun'
AXTE 1.0 'Time [s]'
SCOU  0 'v_0' NPAS 0 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  1 'v_1' NPAS 1 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  2 'v_2' NPAS 2 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  3 'v_3' NPAS 3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  4 'v_4' NPAS 4 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU 10 'l_0' NPAS 0 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 11 'l_1' NPAS 1 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 12 'l_2' NPAS 2 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 13 'l_3' NPAS 3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 14 'l_4' NPAS 4 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
COUR 20 '10-1' SUBC 10 1.0
COUR 21 '11-1' SUBC 11 1.0
COUR 22 '12-1' SUBC 12 1.0
COUR 23 '13-1' SUBC 13 1.0
COUR 24 '14-1' SUBC 14 1.0
COUR 30 '(10-1)/3' DIVC 20 3.0
COUR 31 '(11-1)/3' DIVC 21 3.0
COUR 32 '(12-1)/3' DIVC 22 3.0
COUR 33 '(13-1)/3' DIVC 23 3.0
COUR 34 '(14-1)/3' DIVC 24 3.0
TRAC  0  1  2  3  4 AXES 1.0 'VELOC. [M/S]'
COLO NOIR TURQ VERT ROUG ROSE
TRAC 0 30 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 1 31 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 2 32 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 3 33 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 4 34 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
SCOU  5 'v_0' NPAS 5 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  6 'v_1' NPAS 6 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  7 'v_2' NPAS 7 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  8 'v_3' NPAS 8 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  9 'v_4' NPAS 9 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU 15 'l_0' NPAS 5 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 16 'l_1' NPAS 6 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 17 'l_2' NPAS 7 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 18 'l_3' NPAS 8 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 19 'l_4' NPAS 9 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
COUR 25 '15-1' SUBC 15 1.0
COUR 26 '16-1' SUBC 16 1.0
COUR 27 '17-1' SUBC 17 1.0
COUR 28 '18-1' SUBC 18 1.0
COUR 29 '19-1' SUBC 19 1.0
COUR 35 '(15-1)/3' DIVC 25 3.0
COUR 36 '(16-1)/3' DIVC 26 3.0
COUR 37 '(17-1)/3' DIVC 27 3.0
COUR 38 '(18-1)/3' DIVC 28 3.0
COUR 39 '(19-1)/3' DIVC 29 3.0
TRAC  5  6  7  8  9 AXES 1.0 'VELOC. [M/S]'
COLO NOIR TURQ VERT ROUG ROSE
TRAC 5 35 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 6 36 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 7 37 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 8 38 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 9 39 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
*================================================================
FIN
```

ve1d17e.epx

```
VE1D17C
ECHO
RESU ALIC 've1d17.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT 1 PAS 1 21 TERM
SORT GRAP
PERF 've1d17c.pun'
AXTE 1.0 'Time [s]'
SCOU  0 'v_0' NPAS 10 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  1 'v_1' NPAS 11 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  2 'v_2' NPAS 12 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  3 'v_3' NPAS 13 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  4 'v_4' NPAS 14 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU 10 'l_0' NPAS 10 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 11 'l_1' NPAS 11 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 12 'l_2' NPAS 12 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 13 'l_3' NPAS 13 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 14 'l_4' NPAS 14 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
COUR 20 '10-1' SUBC 10 1.0
COUR 21 '11-1' SUBC 11 1.0
COUR 22 '12-1' SUBC 12 1.0
COUR 23 '13-1' SUBC 13 1.0
COUR 24 '14-1' SUBC 14 1.0
COUR 30 '(10-1)/3' DIVC 20 3.0
COUR 31 '(11-1)/3' DIVC 21 3.0
COUR 32 '(12-1)/3' DIVC 22 3.0
COUR 33 '(13-1)/3' DIVC 23 3.0
COUR 34 '(14-1)/3' DIVC 24 3.0
TRAC  0  1  2  3  4 AXES 1.0 'VELOC. [M/S]'
COLO NOIR TURQ VERT ROUG ROSE
TRAC 0 30 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 1 31 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 2 32 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 3 33 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 4 34 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
SCOU  5 'v_0' NPAS 15 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  6 'v_1' NPAS 16 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  7 'v_2' NPAS 17 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  8 'v_3' NPAS 18 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  9 'v_4' NPAS 19 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU 15 'l_0' NPAS 15 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 16 'l_1' NPAS 16 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 17 'l_2' NPAS 17 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 18 'l_3' NPAS 18 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 19 'l_4' NPAS 19 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
COUR 25 '15-1' SUBC 15 1.0
COUR 26 '16-1' SUBC 16 1.0
COUR 27 '17-1' SUBC 17 1.0
COUR 28 '18-1' SUBC 18 1.0
COUR 29 '19-1' SUBC 19 1.0
COUR 35 '(15-1)/3' DIVC 25 3.0
COUR 36 '(16-1)/3' DIVC 26 3.0
COUR 37 '(17-1)/3' DIVC 27 3.0
COUR 38 '(18-1)/3' DIVC 28 3.0
COUR 39 '(19-1)/3' DIVC 29 3.0
TRAC  5  6  7  8  9 AXES 1.0 'VELOC. [M/S]'
COLO NOIR TURQ VERT ROUG ROSE
TRAC 5 35 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 6 36 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 7 37 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 8 38 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 9 39 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
*================================================================
FIN
```

ve1d18.epx

```
VE1D18
ECHO
!CONV win
LAGR CPLA
DIME
  ADAP NPOI 104 NIND 2 FUN2 208 ENDA
TERM
GEOM LIBR POIN 21 FUN2 20 TERM
 0.00000E+00 0.00000E+00 4.00000E-02 0.00000E+00 8.00000E-02 0.00000E+00
 1.20000E-01 0.00000E+00 1.60000E-01 0.00000E+00 2.00000E-01 0.00000E+00
 2.40000E-01 0.00000E+00 2.80000E-01 0.00000E+00 3.20000E-01 0.00000E+00
 3.60000E-01 0.00000E+00 4.00000E-01 0.00000E+00 4.40000E-01 0.00000E+00
 4.80000E-01 0.00000E+00 5.20000E-01 0.00000E+00 5.60000E-01 0.00000E+00
 6.00000E-01 0.00000E+00 6.40000E-01 0.00000E+00 6.80000E-01 0.00000E+00
 7.20000E-01 0.00000E+00 7.60000E-01 0.00000E+00 8.00000E-01 0.00000E+00
     1     2     2     3     3     4
     4     5     5     6     6     7
     7     8     8     9     9    10
    10    11    11    12    12    13
    13    14    14    15    15    16
    16    17    17    18    18    19
    19    20    20    21
COMP GROU 2 'mesh' LECT 1 PAS 1 20 TERM
            'coro' LECT mesh TERM COND BOX X0 0.3  Y0 -0.02
                                         DX 0.16 DY  0.02
     COUL VERT LECT mesh _fun2 TERM
          ROUG LECT coro TERM
     EPAI 1. LECT mesh _fun2 TERM
ADAP INDI VITE ACCE
     PELE 1000 ALFA 4
     CERR 1.0 LECT mesh TERM
MATE VM23 RO 8000. YOUN 2.E11 NU 0.0 ELAS 2.E11
          TRAC 1 2.E11 1.D0
          LECT mesh _fun2 TE0RM
LINK COUP ! necessaire en adaptivite
INIT ADAP SPLI LEVE 4 LECT coro TERM
     ROUT CASE 3
ECRI COOR DEPL VITE FREQ 1
     FICH ALIC FREQ 1
OPTI NOTE LOG 1
  ADAP STAT RCON
       MAXL 4 DUMP
  LNKS STAT
CALC TINI 0. TEND 40.0E-6
FIN
```

ve1d18a.epx

```
VE1D18A
```

```
ECHO
RESU ALIC 've1d18.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT 1 PAS 1 21 TERM
SORT GRAP
PERF 've1d18a.pun'
AXTE 1.0 'Time [s]'
SCOU  1 'v_01'  T 0.00E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                          VITE COMP 1
SCOU  2 'v_02'  T 0.01E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                          VITE COMP 1
SCOU  3 'v_03'  T 0.02E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                          VITE COMP 1
SCOU  4 'v_04'  T 0.03E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                          VITE COMP 1
SCOU  5 'v_05'  T 0.04E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                          VITE COMP 1
TRAC  1  2  3  4  5 AXES 1.0 'VELOC. [M/S]'
LIST  1  2  3  4  5 AXES 1.0 'VELOC. [M/S]'
*==================================================================
FIN
```

ve1d18acc.epx

```
VE1D18A
ECHO
RESU ALIC 've1d18.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT 1 PAS 1 21 TERM
SORT GRAP
PERF 've1d18a.pun'
AXTE 1.0 'Time [s]'
SCOU  1 'a_01'  T 0.00E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                          ACCE COMP 1
SCOU  2 'a_02'  T 0.01E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                          ACCE COMP 1
SCOU  3 'a_03'  T 0.02E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                          ACCE COMP 1
SCOU  4 'a_04'  T 0.03E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                          ACCE COMP 1
SCOU  5 'a_05'  T 0.04E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                          ACCE COMP 1
TRAC  1  2  3  4  5 AXES 1.0 'ACCE. [M/S2]'
LIST  1  2  3  4  5 AXES 1.0 'ACCE. [M/S2]'
*==================================================================
FIN
```

ve1d18c.epx

```
VE1D18C
ECHO
RESU ALIC 've1d18.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT 1 PAS 1 21 TERM
SORT GRAP
PERF 've1d18c.pun'
AXTE 1.0 'Time [s]'
SCOU  0 'v_0' NPAS 0 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  1 'v_1' NPAS 1 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  2 'v_2' NPAS 2 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  3 'v_3' NPAS 3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  4 'v_4' NPAS 4 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU 10 'l_0' NPAS 0 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 11 'l_1' NPAS 1 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 12 'l_2' NPAS 2 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 13 'l_3' NPAS 3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 14 'l_4' NPAS 4 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
COUR 20 '10-1' SUBC 10 1.0
COUR 21 '11-1' SUBC 11 1.0
COUR 22 '12-1' SUBC 12 1.0
COUR 23 '13-1' SUBC 13 1.0
COUR 24 '14-1' SUBC 14 1.0
COUR 30 '(10-1)/3' DIVC 20 3.0
COUR 31 '(11-1)/3' DIVC 21 3.0
COUR 32 '(12-1)/3' DIVC 22 3.0
COUR 33 '(13-1)/3' DIVC 23 3.0
COUR 34 '(14-1)/3' DIVC 24 3.0
TRAC  0  1  2  3  4 AXES 1.0 'VELOC. [M/S]'
COLO NOIR TURQ VERT ROUG ROSE
TRAC 0 30 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 1 31 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 2 32 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 3 33 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 4 34 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
SCOU 100 'cerr_0' NPAS 0 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     CERR
SCOU 101 'cerr_1' NPAS 1 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     CERR
SCOU 102 'cerr_2' NPAS 2 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     CERR
SCOU 103 'cerr_3' NPAS 3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     CERR
SCOU 104 'cerr_4' NPAS 4 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     CERR
TRAC 100 AXES 1.0 'CERR'
TRAC 101 AXES 1.0 'CERR'
TRAC 102 AXES 1.0 'CERR'
TRAC 103 AXES 1.0 'CERR'
TRAC 104 AXES 1.0 'CERR'
SCOU 110 'maxc_0' NPAS 0 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     MAXC
SCOU 111 'maxc_1' NPAS 1 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     MAXC
SCOU 112 'maxc_2' NPAS 2 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     MAXC
SCOU 113 'maxc_3' NPAS 3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     MAXC
SCOU 114 'maxc_4' NPAS 4 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     MAXC
TRAC 110 AXES 1.0 'MAXC'
TRAC 111 AXES 1.0 'MAXC'
TRAC 112 AXES 1.0 'MAXC'
TRAC 113 AXES 1.0 'MAXC'
TRAC 114 AXES 1.0 'MAXC'
SCOU 120 'erri_0' NPAS 0 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ERRI
SCOU 121 'erri_1' NPAS 1 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ERRI
SCOU 122 'erri_2' NPAS 2 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ERRI
SCOU 123 'erri_3' NPAS 3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ERRI
SCOU 124 'erri_4' NPAS 4 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ERRI
TRAC 120 AXES 1.0 'ERRI'
TRAC 121 AXES 1.0 'ERRI'
TRAC 122 AXES 1.0 'ERRI'
TRAC 123 AXES 1.0 'ERRI'
TRAC 124 AXES 1.0 'ERRI'
SCOU 130 'clen_0' NPAS 0 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     CLEN
SCOU 131 'clen_1' NPAS 1 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     CLEN
SCOU 132 'clen_2' NPAS 2 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     CLEN
SCOU 133 'clen_3' NPAS 3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     CLEN
SCOU 134 'clen_4' NPAS 4 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     CLEN
TRAC 130 AXES 1.0 'CLEN'
TRAC 131 AXES 1.0 'CLEN'
TRAC 132 AXES 1.0 'CLEN'
TRAC 133 AXES 1.0 'CLEN'
TRAC 134 AXES 1.0 'CLEN'
SCOU 140 'ilen_0' NPAS 0 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ILEN
SCOU 141 'ilen_1' NPAS 1 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ILEN
SCOU 142 'ilen_2' NPAS 2 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ILEN
SCOU 143 'ilen_3' NPAS 3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ILEN
SCOU 144 'ilen_4' NPAS 4 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ILEN
TRAC 140 AXES 1.0 'ILEN'
TRAC 141 AXES 1.0 'ILEN'
TRAC 142 AXES 1.0 'ILEN'
TRAC 143 AXES 1.0 'ILEN'
TRAC 144 AXES 1.0 'ILEN'
*==================================================================
FIN
```

ve1d18d.epx

```
VE1D18C
ECHO
RESU ALIC 've1d18.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT 1 PAS 1 21 TERM
SORT GRAP
PERF 've1d18c.pun'
AXTE 1.0 'Time [s]'
SCOU  0 'v_0' NPAS 0 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  1 'v_1' NPAS 1 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  2 'v_2' NPAS 2 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  3 'v_3' NPAS 3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  4 'v_4' NPAS 4 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU 10 'l_0' NPAS 0 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 11 'l_1' NPAS 1 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 12 'l_2' NPAS 2 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 13 'l_3' NPAS 3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 14 'l_4' NPAS 4 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
COUR 20 '10-1' SUBC 10 1.0
COUR 21 '11-1' SUBC 11 1.0
COUR 22 '12-1' SUBC 12 1.0
COUR 23 '13-1' SUBC 13 1.0
COUR 24 '14-1' SUBC 14 1.0
COUR 30 '(10-1)/3' DIVC 20 3.0
COUR 31 '(11-1)/3' DIVC 21 3.0
COUR 32 '(12-1)/3' DIVC 22 3.0
COUR 33 '(13-1)/3' DIVC 23 3.0
COUR 34 '(14-1)/3' DIVC 24 3.0
TRAC  0  1  2  3  4 AXES 1.0 'VELOC. [M/S]'
COLO NOIR TURQ VERT ROUG ROSE
TRAC 0 30 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 1 31 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 2 32 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 3 33 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 4 34 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
SCOU  5 'v_0' NPAS 5 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  6 'v_1' NPAS 6 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  7 'v_2' NPAS 7 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  8 'v_3' NPAS 8 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  9 'v_4' NPAS 9 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU 15 'l_0' NPAS 5 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 16 'l_1' NPAS 6 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
```

```
SCOU 17 'l_2' NPAS 7 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 18 'l_3' NPAS 8 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 19 'l_4' NPAS 9 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
COUR 25 '15-1' SUBC 15 1.0
COUR 26 '16-1' SUBC 16 1.0
COUR 27 '17-1' SUBC 17 1.0
COUR 28 '18-1' SUBC 18 1.0
COUR 29 '19-1' SUBC 19 1.0
COUR 35 '(15-1)/3' DIVC 25 3.0
COUR 36 '(16-1)/3' DIVC 26 3.0
COUR 37 '(17-1)/3' DIVC 27 3.0
COUR 38 '(18-1)/3' DIVC 28 3.0
COUR 39 '(19-1)/3' DIVC 29 3.0
TRAC  5  6  7  8  9 AXES 1.0 'VELOC. [M/S]'
COLO NOIR TURQ VERT ROUG ROSE
TRAC 5 35 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 6 36 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 7 37 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 8 38 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 9 39 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
*==================================================================
FIN
```

ve1d18e.epx

```
VE1D18C
ECHO
RESU ALIC 've1d18.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT 1 PAS 1 21 TERM
SORT GRAP
PERF 've1d18c.pun'
AXTE 1.0 'Time [s]'
SCOU  0 'v_0' NPAS 10 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  1 'v_1' NPAS 11 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  2 'v_2' NPAS 12 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  3 'v_3' NPAS 13 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  4 'v_4' NPAS 14 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU 10 'l_0' NPAS 10 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 11 'l_1' NPAS 11 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 12 'l_2' NPAS 12 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 13 'l_3' NPAS 13 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 14 'l_4' NPAS 14 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
COUR 20 '10-1' SUBC 10 1.0
COUR 21 '11-1' SUBC 11 1.0
COUR 22 '12-1' SUBC 12 1.0
COUR 23 '13-1' SUBC 13 1.0
COUR 24 '14-1' SUBC 14 1.0
COUR 30 '(10-1)/3' DIVC 20 3.0
COUR 31 '(11-1)/3' DIVC 21 3.0
COUR 32 '(12-1)/3' DIVC 22 3.0
COUR 33 '(13-1)/3' DIVC 23 3.0
COUR 34 '(14-1)/3' DIVC 24 3.0
TRAC  0  1  2  3  4 AXES 1.0 'VELOC. [M/S]'
COLO NOIR TURQ VERT ROUG ROSE
TRAC 0 30 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 1 31 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 2 32 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 3 33 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 4 34 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
SCOU  5 'v_0' NPAS 15 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  6 'v_1' NPAS 16 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  7 'v_2' NPAS 17 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  8 'v_3' NPAS 18 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  9 'v_4' NPAS 19 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU 15 'l_0' NPAS 15 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 16 'l_1' NPAS 16 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 17 'l_2' NPAS 17 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 18 'l_3' NPAS 18 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 19 'l_4' NPAS 19 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
COUR 25 '15-1' SUBC 15 1.0
COUR 26 '16-1' SUBC 16 1.0
COUR 27 '17-1' SUBC 17 1.0
COUR 28 '18-1' SUBC 18 1.0
COUR 29 '19-1' SUBC 19 1.0
COUR 35 '(15-1)/3' DIVC 25 3.0
COUR 36 '(16-1)/3' DIVC 26 3.0
COUR 37 '(17-1)/3' DIVC 27 3.0
COUR 38 '(18-1)/3' DIVC 28 3.0
COUR 39 '(19-1)/3' DIVC 29 3.0
TRAC  5  6  7  8  9 AXES 1.0 'VELOC. [M/S]'
COLO NOIR TURQ VERT ROUG ROSE
TRAC 5 35 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 6 36 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 7 37 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 8 38 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
TRAC 9 39 AXES 1.0 'V [M/S] / LEV' COLO NOIR ROUG
*==================================================================
FIN
```

ve1d18eacc.epx

```
VE1D18C
ECHO
RESU ALIC 've1d18.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT 1 PAS 1 21 TERM
SORT GRAP
PERF 've1d18c.pun'
AXTE 1.0 'Time [s]'
SCOU  0 'v_0' NPAS 10 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU  1 'v_1' NPAS 11 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     VITE COMP 1
SCOU 10 'l_0' NPAS 10 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 11 'l_1' NPAS 11 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ETLE
SCOU 40 'a_0' NPAS 10 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ACCE COMP 1
SCOU 41 'a_1' NPAS 11 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                     ACCE COMP 1
COUR 20 '10-1' SUBC 10 1.0
COUR 21 '11-1' SUBC 11 1.0
COUR 30 '(10-1)/3' DIVC 20 3.0
COUR 31 '(11-1)/3' DIVC 21 3.0
COUR 50 'ACC/1E5' DIVC 40 1.E5
COUR 51 'ACC/1E5' DIVC 41 1.E5
TRAC 0 30 50 AXES 1.0 'V/LEV/A' YZER COLO NOIR ROUG VERT
TRAC 1 31 51 AXES 1.0 'V/LEV/A' YZER COLO NOIR ROUG VERT
*==================================================================
FIN
```

ve1d22.epx

```
VE1D22
ECHO
!CONV win
LAGR CPLA
DIME
  ADAP NPOI 815 NIND 2 Q41L 928 ENDA
TERM
GEOM 9 '(6E12.5)' '(12I6)' POIN 42 Q41L 20 TERM
COMP GROU 2 'mesh' LECT 1 PAS 1 20 TERM
            'coro' LECT mesh TERM COND BOX X0 0.3 Y0 0 DX 0.16 DY 0.04
     COUL VERT LECT mesh _q41l TERM
          ROUG LECT coro TERM
     EPAI 1. LECT mesh _q41l TERM
ADAP INDI VITE ACCE
     PELE 1000 ALFA 4
     CERR 1.0 LECT mesh TERM
MATE VM23 RO 8000. YOUN 2.E11 NU 0.0 ELAS 2.E11
          TRAC 1 2.E11 1.D0
          LECT mesh _q41l TE0RM
LINK COUP ! necessaire en adaptivite
INIT ADAP SPLI LEVE 4 LECT coro TERM
     ROUT CASE 3
ECRI COOR DEPL VITE ACCE FINT FEXT CONT ECRO TFRE 40.E-6
     FICH ALIC FREQ 1
OPTI NOTE LOG 1
  ADAP STAT RCON MAXL 4 dump
  LNKS STAT
CALC TINI 0. TEND 40.0E-6
FIN
```

ve1d22b.epx

```
VE1D22B
ECHO
RESU ALIC 've1d22.ali' GARD PSCR
SORT VISU NSTO 1
*=====================================================================
PLAY
CAME   1 EYE   4.00000E-01  2.00000E-02  2.00250E+00
!        Q     1.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
         VIEW  0.00000E+00  0.00000E+00 -1.00000E+00
         RIGH  1.00000E+00  0.00000E+00  0.00000E+00
         UP    0.00000E+00  1.00000E+00  0.00000E+00
         FOV   2.48819E+01
SCEN GEOM NAVI FREE
     FACE HFRO
!    LINE HEOU !SFRE
!    VECT SCCO FIEL VITE SCAL USER PROG 0.07 PAS 0.07 0.98
!    TEXT VSCA
!    ISO  FILL FIEL VITE 1 SCAL USER PROG -0.975 PAS 0.15 0.975
!    ISO  FILL FIEL VITE 2 SCAL USER PROG -0.975 PAS 0.15 0.975
!    TEXT ISCA
     COLO PAPE
SLER CAM1 1 NFRA 1
TRAC OFFS FICH AVI NOCL NFTO 51 FPS 15 KFRE 10 COMP -1 REND
FREQ 1
GOTR LOOP 49 OFFS FICH AVI CONT NOCL REND
GO
TRAC OFFS FICH AVI CONT REND
ENDPLAY
*=====================================================================
FIN
```

ve1d22c.epx

```
VE1D22C
ECHO
RESU ALIC 've1d22.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT 1 PAS 1 21 TERM
SORT GRAP
PERF 've1d22c.pun'
AXTE 1.0 'Time [s]'
SCOU  1 'v_01'  T 0.00E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          VITE COMP 1
SCOU  2 'v_02'  T 0.01E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
```

```
                         VITE COMP 1
SCOU  3 'v_03'  T 0.02E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                         VITE COMP 1
SCOU  4 'v_04'  T 0.03E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                         VITE COMP 1
SCOU  5 'v_05'  T 0.04E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                         VITE COMP 1
RCOU 101 'v_01' FICH 've1d01c.pun' RENA 'v_01_01'
RCOU 102 'v_02' FICH 've1d01c.pun' RENA 'v_02_01'
RCOU 103 'v_03' FICH 've1d01c.pun' RENA 'v_03_01'
RCOU 104 'v_04' FICH 've1d01c.pun' RENA 'v_04_01'
RCOU 105 'v_05' FICH 've1d01c.pun' RENA 'v_05_01'
TRAC  1  2  3  4  5 AXES 1.0 'VELOC. [M/S]'
LIST  1  2  3  4  5 AXES 1.0 'VELOC. [M/S]'
TRAC  1 5 AXES 1.0 'VELOC. [M/S]'
COLO NOIR ROUG
TRAC  1 101 AXES 1.0 'VELOC. [M/S]'
COLO NOIR ROUG
TRAC  2 102 AXES 1.0 'VELOC. [M/S]'
COLO NOIR ROUG
TRAC  3 103 AXES 1.0 'VELOC. [M/S]'
COLO NOIR ROUG
TRAC  4 104 AXES 1.0 'VELOC. [M/S]'
COLO NOIR ROUG
TRAC  5 105 AXES 1.0 'VELOC. [M/S]'
COLO NOIR ROUG
SCOU 11 'el_01'  T 0.00E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          ETLE
SCOU 12 'el_02'  T 0.01E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          ETLE
SCOU 13 'el_03'  T 0.02E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          ETLE
SCOU 14 'el_04'  T 0.03E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          ETLE
SCOU 15 'el_05'  T 0.04E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          ETLE
TRAC 11 12 13 14 15 AXES 1.0 'ETLE [-]'
LIST 11 12 13 14 15 AXES 1.0 'ETLE [-]'
TRAC  1 11 AXES 1.0 'V&ETLE'
COLO NOIR VERT
TRAC  2 12 AXES 1.0 'V&ETLE'
COLO NOIR VERT
TRAC  3 13 AXES 1.0 'V&ETLE'
COLO NOIR VERT
TRAC  4 14 AXES 1.0 'V&ETLE'
COLO NOIR VERT
TRAC  5 15 AXES 1.0 'V&ETLE'
COLO NOIR VERT
COUR 21 '11-1' SUBC 11 1.0
COUR 22 '12-1' SUBC 12 1.0
COUR 23 '13-1' SUBC 13 1.0
COUR 24 '14-1' SUBC 14 1.0
COUR 25 '15-1' SUBC 15 1.0
COUR 31 '(11-1)/3' DIVC 21 3.0
COUR 32 '(12-1)/6' DIVC 22 6.0
COUR 33 '(13-1)/6' DIVC 23 6.0
COUR 34 '(14-1)/6' DIVC 24 6.0
COUR 35 '(15-1)/6' DIVC 25 6.0
TRAC  1 101 31 AXES 1.0 'V&ETLE_NOR'
COLO NOIR ROUG VERT
TRAC  2 102 32 AXES 1.0 'V&ETLE_NOR'
COLO NOIR ROUG VERT
TRAC  3 103 33 AXES 1.0 'V&ETLE_NOR'
COLO NOIR ROUG VERT
TRAC  4 104 34 AXES 1.0 'V&ETLE_NOR'
COLO NOIR ROUG VERT
TRAC  5 105 35 AXES 1.0 'V&ETLE_NOR'
COLO NOIR ROUG VERT
SCOU 61 'nl_01'  T 0.00E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          NTLE
SCOU 62 'nl_02'  T 0.01E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          NTLE
SCOU 63 'nl_03'  T 0.02E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          NTLE
SCOU 64 'nl_04'  T 0.03E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          NTLE
SCOU 65 'nl_05'  T 0.04E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          NTLE
TRAC 61 62 63 64 65 AXES 1.0 'NTLE [-]'
LIST 61 62 63 64 65 AXES 1.0 'NTLE [-]'
TRAC 11 61 AXES 1.0 'ETLE&NTLE'
COLO NOIR VERT
TRAC 12 62 AXES 1.0 'ETLE&NTLE'
COLO NOIR VERT
TRAC 13 63 AXES 1.0 'ETLE&NTLE'
COLO NOIR VERT
TRAC 14 64 AXES 1.0 'ETLE&NTLE'
COLO NOIR VERT
TRAC 15 65 AXES 1.0 'ETLE&NTLE'
COLO NOIR VERT
*==================================================================
FIN
```

ve1d22c1.epx

```
VE1D22C
ECHO
RESU ALIC 've1d22.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT 1 PAS 1 21 TERM
SORT GRAP
PERF 've1d22c1.pun'
AXTE 1.0 'Time [s]'
SCOU 100 'v_00'  NPAS  0 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                         VITE COMP 1
SCOU 101 'v_01'  NPAS  1 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                         VITE COMP 1
SCOU 102 'v_02'  NPAS  2 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                         VITE COMP 1
SCOU 103 'v_03'  NPAS  3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                         VITE COMP 1
SCOU 104 'v_04'  NPAS  4 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                         VITE COMP 1
SCOU 105 'v_05'  NPAS  5 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                         VITE COMP 1
SCOU 106 'v_06'  NPAS  6 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                         VITE COMP 1
SCOU 107 'v_07'  NPAS  7 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                         VITE COMP 1
SCOU 108 'v_08'  NPAS  8 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                         VITE COMP 1
SCOU 109 'v_09'  NPAS  9 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                         VITE COMP 1
SCOU 110 'v_10'  NPAS 10 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                         VITE COMP 1
SCOU 111 'v_11'  NPAS 11 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                         VITE COMP 1
SCOU 112 'v_12'  NPAS 12 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                         VITE COMP 1
SCOU 113 'v_13'  NPAS 13 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                         VITE COMP 1
TRAC 100 101 102 103 104 105 106 AXES 1.0 'VELOC. [M/S]'
COLO NOIR BLEU TURQ VERT JAUN ROSE ROUG
TRAC 107 108 109 110 111 112 113 AXES 1.0 'VELOC. [M/S]'
COLO NOIR BLEU TURQ VERT JAUN ROSE ROUG
LIST 100 101 102 103 104 105 106 AXES 1.0 'VELOC. [M/S]'
LIST 107 108 109 110 111 112 113 AXES 1.0 'VELOC. [M/S]'
SCOU 200 'el_00'  NPAS  0 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          ETLE
SCOU 201 'el_01'  NPAS  1 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          ETLE
SCOU 202 'el_02'  NPAS  2 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          ETLE
SCOU 203 'el_03'  NPAS  3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          ETLE
SCOU 204 'el_04'  NPAS  4 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          ETLE
SCOU 205 'el_05'  NPAS  5 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          ETLE
SCOU 206 'el_06'  NPAS  6 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          ETLE
SCOU 207 'el_07'  NPAS  7 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          ETLE
SCOU 208 'el_08'  NPAS  8 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          ETLE
SCOU 209 'el_09'  NPAS  9 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          ETLE
SCOU 210 'el_10'  NPAS 10 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          ETLE
SCOU 211 'el_11'  NPAS 11 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          ETLE
SCOU 212 'el_12'  NPAS 12 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          ETLE
SCOU 213 'el_13'  NPAS 13 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          ETLE
TRAC 200 201 202 203 204 205 206 AXES 1.0 'ETLE [-]'
COLO NOIR BLEU TURQ VERT JAUN ROSE ROUG
TRAC 207 208 209 210 211 212 213 AXES 1.0 'ETLE [-]'
COLO NOIR BLEU TURQ VERT JAUN ROSE ROUG
LIST 200 201 202 203 204 205 206 AXES 1.0 'ETLE [-]'
LIST 207 208 209 210 211 212 213 AXES 1.0 'ETLE [-]'
TRAC 100 200 AXES 1.0 'V&ETLE'
COLO NOIR VERT
TRAC 100 200 AXES 1.0 'V&ETLE'
COLO NOIR VERT
TRAC 101 201 AXES 1.0 'V&ETLE'
COLO NOIR VERT
TRAC 102 202 AXES 1.0 'V&ETLE'
COLO NOIR VERT
TRAC 103 203 AXES 1.0 'V&ETLE'
COLO NOIR VERT
TRAC 104 204 AXES 1.0 'V&ETLE'
COLO NOIR VERT
TRAC 105 205 AXES 1.0 'V&ETLE'
COLO NOIR VERT
TRAC 106 206 AXES 1.0 'V&ETLE'
COLO NOIR VERT
TRAC 107 207 AXES 1.0 'V&ETLE'
COLO NOIR VERT
TRAC 108 208 AXES 1.0 'V&ETLE'
COLO NOIR VERT
TRAC 109 209 AXES 1.0 'V&ETLE'
COLO NOIR VERT
TRAC 110 210 AXES 1.0 'V&ETLE'
COLO NOIR VERT
TRAC 111 211 AXES 1.0 'V&ETLE'
COLO NOIR VERT
TRAC 112 212 AXES 1.0 'V&ETLE'
COLO NOIR VERT
TRAC 113 213 AXES 1.0 'V&ETLE'
COLO NOIR VERT
*==================================================================
FIN
```

ve1d22cl.epx

```
VE1D22CL
ECHO
RESU ALIC 've1d22.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT 1 PAS 1 21 TERM
SORT GRAP
PERF 've1d22cl.pun'
AXTE 1.0 'Time [s]'
SCOU 11 'c_01'  T 0.00E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          CLEN
SCOU 12 'c_02'  T 0.01E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          CLEN
SCOU 13 'c_03'  T 0.02E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          CLEN
SCOU 14 'c_04'  T 0.03E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          CLEN
SCOU 15 'c_05'  T 0.04E-3 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
                          CLEN
TRAC 11 12 13 14 15 AXES 1.0 'CLEN [M]'
LIST 11 12 13 14 15 AXES 1.0 'CLEN [M]'
*==================================================================
FIN
```

ve1d22t.epx

```
VE1D22T
ECHO
RESU ALIC 've1d22.ali' GARD PSCR
SORT GRAP
PERF 've1d22t.pun'
AXTE 1.0 'Time [s]'
COUR   1 'dt1'  DT1
COUR 101 'nspl' NSPL
COUR 102 'nusp' NUSP
COUR 103 'nspt' NSPT
COUR 104 'nust' NUST
COUR 105 'nuse' NUSE
COUR 106 'nact' NACT
COUR 107 'nusn' NUSN
COUR 108 'lmax' LMAX
COUR 109 'lmin' LMIN
TRAC   1 AXES 1.0 'DT1 [S]'
TRAC 101 AXES 1.0 'SPLIT ELEMENTS'
TRAC 102 AXES 1.0 'UNSPLIT ELEMENTS'
TRAC 103 AXES 1.0 'TOT SPLIT ELEMS'
TRAC 104 AXES 1.0 'TOT UNSPLIT ELS'
TRAC 105 AXES 1.0 'USED ELEMENTS'
TRAC 106 AXES 1.0 'ACTIVE ELEMENTS'
TRAC 107 AXES 1.0 'USED NODES'
TRAC 108 AXES 1.0 'MAX LEVEL'
TRAC 109 AXES 1.0 'MIN LEVEL'
TRAC 103 104 AXES 1.0 'TOT SP/USP ELS'
COLO NOIR ROUG
TRAC 105 106 AXES 1.0 'USED/ACTIVE ELS'
COLO NOIR ROUG
TRAC 108 109 AXES 1.0 'MAX/MIN LEVELS'
COLO NOIR ROUG
LIST   1 AXES 1.0 'DT1 [S]'
LIST 101 AXES 1.0 'SPLIT ELEMENTS'
LIST 102 AXES 1.0 'UNSPLIT ELEMENTS'
LIST 103 AXES 1.0 'TOT SPLIT ELEMS'
LIST 104 AXES 1.0 'TOT UNSPLIT ELS'
LIST 105 AXES 1.0 'USED ELEMENTS'
LIST 106 AXES 1.0 'ACTIVE ELEMENTS'
LIST 107 AXES 1.0 'USED NODES'
LIST 108 AXES 1.0 'MAX LEVEL'
LIST 109 AXES 1.0 'MIN LEVEL'
*=====================================================================
FIN
```

ve1d27.epx

```
VE1D27
ECHO
!CONV win
LAGR CPLA
GEOM LIBR POIN 161 FUN2 160 TERM
 0.0000  0.0
 0.0050  0.0
 0.0100  0.0
 0.0150  0.0
 0.0200  0.0
 0.0250  0.0
 0.0300  0.0
 0.0350  0.0
 0.0400  0.0
 0.0450  0.0
 0.0500  0.0
 0.0550  0.0
 0.0600  0.0
 0.0650  0.0
 0.0700  0.0
 0.0750  0.0
 0.0800  0.0
 0.0850  0.0
 0.0900  0.0
 0.0950  0.0
 0.1000  0.0
 0.1050  0.0
 0.1100  0.0
 0.1150  0.0
 0.1200  0.0
 0.1250  0.0
 0.1300  0.0
 0.1350  0.0
 0.1400  0.0
 0.1450  0.0
 0.1500  0.0
 0.1550  0.0
 0.1600  0.0
 0.1650  0.0
 0.1700  0.0
 0.1750  0.0
 0.1800  0.0
 0.1850  0.0
 0.1900  0.0
 0.1950  0.0
 0.2000  0.0
 0.2050  0.0
 0.2100  0.0
 0.2150  0.0
 0.2200  0.0
 0.2250  0.0
 0.2300  0.0
 0.2350  0.0
 0.2400  0.0
 0.2450  0.0
 0.2500  0.0
 0.2550  0.0
 0.2600  0.0
 0.2650  0.0
 0.2700  0.0
 0.2750  0.0
 0.2800  0.0
 0.2850  0.0
 0.2900  0.0
 0.2950  0.0
 0.3000  0.0
 0.3050  0.0
 0.3100  0.0
 0.3150  0.0
 0.3200  0.0
 0.3250  0.0
 0.3300  0.0
 0.3350  0.0
 0.3400  0.0
 0.3450  0.0
 0.3500  0.0
 0.3550  0.0
 0.3600  0.0
 0.3650  0.0
 0.3700  0.0
 0.3750  0.0
 0.3800  0.0
 0.3850  0.0
 0.3900  0.0
 0.3950  0.0
 0.4000  0.0
 0.4050  0.0
 0.4100  0.0
 0.4150  0.0
 0.4200  0.0
 0.4250  0.0
 0.4300  0.0
 0.4350  0.0
 0.4400  0.0
 0.4450  0.0
 0.4500  0.0
 0.4550  0.0
 0.4600  0.0
 0.4650  0.0
 0.4700  0.0
 0.4750  0.0
 0.4800  0.0
 0.4850  0.0
 0.4900  0.0
 0.4950  0.0
 0.5000  0.0
 0.5050  0.0
 0.5100  0.0
 0.5150  0.0
 0.5200  0.0
 0.5250  0.0
 0.5300  0.0
 0.5350  0.0
 0.5400  0.0
 0.5450  0.0
 0.5500  0.0
 0.5550  0.0
 0.5600  0.0
 0.5650  0.0
 0.5700  0.0
 0.5750  0.0
 0.5800  0.0
 0.5850  0.0
 0.5900  0.0
 0.5950  0.0
 0.6000  0.0
 0.6050  0.0
 0.6100  0.0
 0.6150  0.0
 0.6200  0.0
 0.6250  0.0
 0.6300  0.0
 0.6350  0.0
 0.6400  0.0
 0.6450  0.0
 0.6500  0.0
 0.6550  0.0
 0.6600  0.0
 0.6650  0.0
 0.6700  0.0
 0.6750  0.0
 0.6800  0.0
 0.6850  0.0
 0.6900  0.0
 0.6950  0.0
 0.7000  0.0
 0.7050  0.0
 0.7100  0.0
 0.7150  0.0
 0.7200  0.0
 0.7250  0.0
 0.7300  0.0
 0.7350  0.0
 0.7400  0.0
 0.7450  0.0
 0.7500  0.0
 0.7550  0.0
 0.7600  0.0
 0.7650  0.0
 0.7700  0.0
 0.7750  0.0
 0.7800  0.0
 0.7850  0.0
 0.7900  0.0
 0.7950  0.0
 0.8000  0.0
   1   2
   2   3
   3   4
   4   5
   5   6
   6   7
   7   8
   8   9
   9  10
  10  11
  11  12
  12  13
  13  14
  14  15
  15  16
  16  17
  17  18
```

```
 18  19
 19  20
 20  21
 21  22
 22  23
 23  24
 24  25
 25  26
 26  27
 27  28
 28  29
 29  30
 30  31
 31  32
 32  33
 33  34
 34  35
 35  36
 36  37
 37  38
 38  39
 39  40
 40  41
 41  42
 42  43
 43  44
 44  45
 45  46
 46  47
 47  48
 48  49
 49  50
 50  51
 51  52
 52  53
 53  54
 54  55
 55  56
 56  57
 57  58
 58  59
 59  60
 60  61
 61  62
 62  63
 63  64
 64  65
 65  66
 66  67
 67  68
 68  69
 69  70
 70  71
 71  72
 72  73
 73  74
 74  75
 75  76
 76  77
 77  78
 78  79
 79  80
 80  81
 81  82
 82  83
 83  84
 84  85
 85  86
 86  87
 87  88
 88  89
 89  90
 90  91
 91  92
 92  93
 93  94
 94  95
 95  96
 96  97
 97  98
 98  99
 99 100
100 101
101 102
102 103
103 104
104 105
105 106
106 107
107 108
108 109
109 110
110 111
111 112
112 113
113 114
114 115
115 116
116 117
117 118
118 119
119 120
120 121
121 122
122 123
123 124
124 125
125 126
126 127
127 128
128 129
129 130
130 131
131 132
132 133
133 134
134 135
135 136
 136 137
 137 138
 138 139
 139 140
 140 141
 141 142
 142 143
 143 144
 144 145
 145 146
 146 147
 147 148
 148 149
 149 150
 150 151
 151 152
 152 153
 153 154
 154 155
 155 156
 156 157
 157 158
 158 159
 159 160
 160 161
COMP GROU 1 'mesh' LECT 1 PAS 1 160 TERM
     EPAI 1. LECT mesh TERM
MATE VM23 RO 8000. YOUN 2.E11 NU 0.0 ELAS 2.E11
          TRAC 1 2.E11 1.D0
          LECT mesh TERM
INIT VITE LIST
     1 0.00000E+00 0.00000E+00     2 0.00000E+00 0.00000E+00
     3 0.00000E+00 0.00000E+00     4 0.00000E+00 0.00000E+00
     5 0.00000E+00 0.00000E+00     6 0.00000E+00 0.00000E+00
     7 0.00000E+00 0.00000E+00     8 0.00000E+00 0.00000E+00
     9 0.00000E+00 0.00000E+00    10 0.00000E+00 0.00000E+00
    11 0.00000E+00 0.00000E+00    12 0.00000E+00 0.00000E+00
    13 0.00000E+00 0.00000E+00    14 0.00000E+00 0.00000E+00
    15 0.00000E+00 0.00000E+00    16 0.00000E+00 0.00000E+00
    17 0.00000E+00 0.00000E+00    18 0.00000E+00 0.00000E+00
    19 0.00000E+00 0.00000E+00    20 0.00000E+00 0.00000E+00
    21 0.00000E+00 0.00000E+00    22 0.00000E+00 0.00000E+00
    23 0.00000E+00 0.00000E+00    24 0.00000E+00 0.00000E+00
    25 0.00000E+00 0.00000E+00    26 0.00000E+00 0.00000E+00
    27 0.00000E+00 0.00000E+00    28 0.00000E+00 0.00000E+00
    29 0.00000E+00 0.00000E+00    30 0.00000E+00 0.00000E+00
    31 0.00000E+00 0.00000E+00    32 0.00000E+00 0.00000E+00
    33 0.00000E+00 0.00000E+00    34 0.00000E+00 0.00000E+00
    35 0.00000E+00 0.00000E+00    36 0.00000E+00 0.00000E+00
    37 0.00000E+00 0.00000E+00    38 0.00000E+00 0.00000E+00
    39 0.00000E+00 0.00000E+00    40 0.00000E+00 0.00000E+00
    41 0.00000E+00 0.00000E+00    42 0.00000E+00 0.00000E+00
    43 0.00000E+00 0.00000E+00    44 0.00000E+00 0.00000E+00
    45 0.00000E+00 0.00000E+00    46 0.00000E+00 0.00000E+00
    47 0.00000E+00 0.00000E+00    48 0.00000E+00 0.00000E+00
    49 0.00000E+00 0.00000E+00    50 0.00000E+00 0.00000E+00
    51 0.00000E+00 0.00000E+00    52 0.00000E+00 0.00000E+00
    53 0.00000E+00 0.00000E+00    54 0.00000E+00 0.00000E+00
    55 0.00000E+00 0.00000E+00    56 0.00000E+00 0.00000E+00
    57 0.00000E+00 0.00000E+00    58 0.00000E+00 0.00000E+00
    59 0.00000E+00 0.00000E+00    60 0.00000E+00 0.00000E+00
    61 0.00000E+00 0.00000E+00    62 0.00000E+00 0.00000E+00
    63 0.00000E+00 0.00000E+00    64 0.00000E+00 0.00000E+00
    65 0.00000E+00 0.00000E+00    66 0.11230E-01 0.00000E+00
    67 0.42969E-01 0.00000E+00    68 0.92285E-01 0.00000E+00
    69 0.15625E+00 0.00000E+00    70 0.23193E+00 0.00000E+00
    71 0.31641E+00 0.00000E+00    72 0.40674E+00 0.00000E+00
    73 0.50000E+00 0.00000E+00    74 0.59326E+00 0.00000E+00
    75 0.68359E+00 0.00000E+00    76 0.76807E+00 0.00000E+00
    77 0.84375E+00 0.00000E+00    78 0.90771E+00 0.00000E+00
    79 0.95703E+00 0.00000E+00    80 0.98877E+00 0.00000E+00
    81 0.10000E+01 0.00000E+00    82 0.98877E+00 0.00000E+00
    83 0.95703E+00 0.00000E+00    84 0.90771E+00 0.00000E+00
    85 0.84375E+00 0.00000E+00    86 0.76807E+00 0.00000E+00
    87 0.68359E+00 0.00000E+00    88 0.59326E+00 0.00000E+00
    89 0.50000E+00 0.00000E+00    90 0.40674E+00 0.00000E+00
    91 0.31641E+00 0.00000E+00    92 0.23193E+00 0.00000E+00
    93 0.15625E+00 0.00000E+00    94 0.92285E-01 0.00000E+00
    95 0.42969E-01 0.00000E+00    96 0.11230E-01 0.00000E+00
    97 0.13312E-29 0.00000E+00    98 0.00000E+00 0.00000E+00
    99 0.00000E+00 0.00000E+00   100 0.00000E+00 0.00000E+00
   101 0.00000E+00 0.00000E+00   102 0.00000E+00 0.00000E+00
   103 0.00000E+00 0.00000E+00   104 0.00000E+00 0.00000E+00
   105 0.00000E+00 0.00000E+00   106 0.00000E+00 0.00000E+00
   107 0.00000E+00 0.00000E+00   108 0.00000E+00 0.00000E+00
   109 0.00000E+00 0.00000E+00   110 0.00000E+00 0.00000E+00
   111 0.00000E+00 0.00000E+00   112 0.00000E+00 0.00000E+00
   113 0.00000E+00 0.00000E+00   114 0.00000E+00 0.00000E+00
   115 0.00000E+00 0.00000E+00   116 0.00000E+00 0.00000E+00
   117 0.00000E+00 0.00000E+00   118 0.00000E+00 0.00000E+00
   119 0.00000E+00 0.00000E+00   120 0.00000E+00 0.00000E+00
   121 0.00000E+00 0.00000E+00   122 0.00000E+00 0.00000E+00
   123 0.00000E+00 0.00000E+00   124 0.00000E+00 0.00000E+00
   125 0.00000E+00 0.00000E+00   126 0.00000E+00 0.00000E+00
   127 0.00000E+00 0.00000E+00   128 0.00000E+00 0.00000E+00
   129 0.00000E+00 0.00000E+00   130 0.00000E+00 0.00000E+00
   131 0.00000E+00 0.00000E+00   132 0.00000E+00 0.00000E+00
   133 0.00000E+00 0.00000E+00   134 0.00000E+00 0.00000E+00
   135 0.00000E+00 0.00000E+00   136 0.00000E+00 0.00000E+00
   137 0.00000E+00 0.00000E+00   138 0.00000E+00 0.00000E+00
   139 0.00000E+00 0.00000E+00   140 0.00000E+00 0.00000E+00
   141 0.00000E+00 0.00000E+00   142 0.00000E+00 0.00000E+00
   143 0.00000E+00 0.00000E+00   144 0.00000E+00 0.00000E+00
   145 0.00000E+00 0.00000E+00   146 0.00000E+00 0.00000E+00
   147 0.00000E+00 0.00000E+00   148 0.00000E+00 0.00000E+00
   149 0.00000E+00 0.00000E+00   150 0.00000E+00 0.00000E+00
   151 0.00000E+00 0.00000E+00   152 0.00000E+00 0.00000E+00
   153 0.00000E+00 0.00000E+00   154 0.00000E+00 0.00000E+00
   155 0.00000E+00 0.00000E+00   156 0.00000E+00 0.00000E+00
   157 0.00000E+00 0.00000E+00   158 0.00000E+00 0.00000E+00
   159 0.00000E+00 0.00000E+00   160 0.00000E+00 0.00000E+00
   161 0.00000E+00 0.00000E+00
ECRI COOR DEPL VITE FREQ 1
     FICH ALIC FREQ 1
OPTI NOTE LOG 1
CALC TINI 0. TEND 40.0E-6
*=================================================================
SUIT
```

```
VE1D27A
ECHO
RESU ALIC GARD PSCR
COMP NGRO 1 'xaxo' LECT 1 PAS 1 161 TERM
SORT GRAP
PERF 've1d27a.pun'
AXTE 1.0 'Time [s]'
SCOU  1 'v_01'  T 0.00E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         VITE COMP 1
SCOU  2 'v_02'  T 0.01E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         VITE COMP 1
SCOU  3 'v_03'  T 0.02E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         VITE COMP 1
SCOU  4 'v_04'  T 0.03E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         VITE COMP 1
SCOU  5 'v_05'  T 0.04E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         VITE COMP 1
TRAC  1  2  3  4  5 AXES 1.0 'VELOC. [M/S]'
LIST  1  2  3  4  5 AXES 1.0 'VELOC. [M/S]'
QUAL VITE COMP 1 LECT  41 TERM REFE  4.96787E-1 TOLE 1.E-2
     VITE COMP 1 LECT 121 TERM REFE  4.96760E-1 TOLE 1.E-2
*=============================================================
SUIT
VE1D17
ECHO
!CONV win
LAGR CPLA
DIME
  ADAP NPOI 104 FUN2 208 ENDA
TERM
GEOM LIBR POIN 21 FUN2 20 TERM
 0.00000E+00 0.00000E+00 4.00000E-02 0.00000E+00 8.00000E-02 0.00000E+00
 1.20000E-01 0.00000E+00 1.60000E-01 0.00000E+00 2.00000E-01 0.00000E+00
 2.40000E-01 0.00000E+00 2.80000E-01 0.00000E+00 3.20000E-01 0.00000E+00
 3.60000E-01 0.00000E+00 4.00000E-01 0.00000E+00 4.40000E-01 0.00000E+00
 4.80000E-01 0.00000E+00 5.20000E-01 0.00000E+00 5.60000E-01 0.00000E+00
 6.00000E-01 0.00000E+00 6.40000E-01 0.00000E+00 6.80000E-01 0.00000E+00
 7.20000E-01 0.00000E+00 7.60000E-01 0.00000E+00 8.00000E-01 0.00000E+00
     1     2     2     3     3     4
     4     5     5     6     6     7
     7     8     8     9     9    10
    10    11    11    12    12    13
    13    14    14    15    15    16
    16    17    17    18    18    19
    19    20    20    21
COMP GROU 2 'mesh' LECT 1 PAS 1 20 TERM
            'coro' LECT mesh TERM COND BOX X0 0.3  Y0 -0.02
                                          DX 0.16 DY  0.02
     COUL VERT LECT mesh _fun2 TERM
          ROUG LECT coro TERM
     EPAI 1. LECT mesh _fun2 TERM
ADAP INDI VITE
     PELE 1000 ALFA 4
     CERR 1.0 LECT mesh TERM
MATE VM23 RO 8000. YOUN 2.E11 NU 0.0 ELAS 2.E11
          TRAC 1 2.E11 1.D0
          LECT mesh _fun2 TE0RM
LINK COUP ! necessaire en adaptivite
INIT ADAP SPLI LEVE 4 LECT coro TERM
     ROUT CASE 3
ECRI COOR DEPL VITE FREQ 1
     FICH ALIC FREQ 1
OPTI NOTE LOG 1
  ADAP STAT RCON
       MAXL 4 DUMP
CALC TINI 0. TEND 40.0E-6
*=============================================================
SUIT
VE1D17A
ECHO
RESU ALIC GARD PSCR
COMP NGRO 1 'xaxo' LECT 1 PAS 1 21 TERM
SORT GRAP
PERF 've1d17a.pun'
AXTE 1.0 'Time [s]'
SCOU  1 'v_01'  T 0.00E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         VITE COMP 1
SCOU  2 'v_02'  T 0.01E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         VITE COMP 1
SCOU  3 'v_03'  T 0.02E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         VITE COMP 1
SCOU  4 'v_04'  T 0.03E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         VITE COMP 1
SCOU  5 'v_05'  T 0.04E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         VITE COMP 1
TRAC  1  2  3  4  5 AXES 1.0 'VELOC. [M/S]'
LIST  1  2  3  4  5 AXES 1.0 'VELOC. [M/S]'
RCOU 11 'v_01' FICH 've1d27a.pun' RENA 'v_01_27'
RCOU 12 'v_02' FICH 've1d27a.pun' RENA 'v_02_27'
RCOU 13 'v_03' FICH 've1d27a.pun' RENA 'v_03_27'
RCOU 14 'v_04' FICH 've1d27a.pun' RENA 'v_04_27'
RCOU 15 'v_05' FICH 've1d27a.pun' RENA 'v_05_27'
TRAC  1  2  3  4  5 11 12 13 14 15 AXES 1.0 'VELOC. [M/S]'
COLO NOIR NOIR NOIR NOIR NOIR
     ROUG ROUG ROUG ROUG ROUG
QUAL VITE COMP 1 LECT  6 TERM REFE  4.93596E-1 TOLE 1.E-2
     VITE COMP 1 LECT 16 TERM REFE  4.93591E-1 TOLE 1.E-2
*=============================================================
SUIT
VE1D18
ECHO
!CONV win
LAGR CPLA
DIME
  ADAP NPOI 104 NIND 2 FUN2 208 ENDA
TERM
GEOM LIBR POIN 21 FUN2 20 TERM
 0.00000E+00 0.00000E+00 4.00000E-02 0.00000E+00 8.00000E-02 0.00000E+00
 1.20000E-01 0.00000E+00 1.60000E-01 0.00000E+00 2.00000E-01 0.00000E+00
 2.40000E-01 0.00000E+00 2.80000E-01 0.00000E+00 3.20000E-01 0.00000E+00
 3.60000E-01 0.00000E+00 4.00000E-01 0.00000E+00 4.40000E-01 0.00000E+00
 4.80000E-01 0.00000E+00 5.20000E-01 0.00000E+00 5.60000E-01 0.00000E+00
 6.00000E-01 0.00000E+00 6.40000E-01 0.00000E+00 6.80000E-01 0.00000E+00
 7.20000E-01 0.00000E+00 7.60000E-01 0.00000E+00 8.00000E-01 0.00000E+00
     1     2     2     3     3     4
     4     5     5     6     6     7
     7     8     8     9     9    10
    10    11    11    12    12    13
    13    14    14    15    15    16
    16    17    17    18    18    19
    19    20    20    21
COMP GROU 2 'mesh' LECT 1 PAS 1 20 TERM
            'coro' LECT mesh TERM COND BOX X0 0.3  Y0 -0.02
                                          DX 0.16 DY  0.02
     COUL VERT LECT mesh _fun2 TERM
          ROUG LECT coro TERM
     EPAI 1. LECT mesh _fun2 TERM
ADAP INDI VITE ACCE
     PELE 1000 ALFA 4
     CERR 1.0 LECT mesh TERM
MATE VM23 RO 8000. YOUN 2.E11 NU 0.0 ELAS 2.E11
          TRAC 1 2.E11 1.D0
          LECT mesh _fun2 TE0RM
LINK COUP ! necessaire en adaptivite
INIT ADAP SPLI LEVE 4 LECT coro TERM
     ROUT CASE 3
ECRI COOR DEPL VITE FREQ 1
     FICH ALIC FREQ 1
OPTI NOTE LOG 1
  ADAP STAT RCON
       MAXL 4 DUMP
CALC TINI 0. TEND 40.0E-6
*=============================================================
SUIT
VE1D17A
ECHO
RESU ALIC GARD PSCR
COMP NGRO 1 'xaxo' LECT 1 PAS 1 21 TERM
SORT GRAP
PERF 've1d18a.pun'
AXTE 1.0 'Time [s]'
SCOU  1 'v_01'  T 0.00E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         VITE COMP 1
SCOU  2 'v_02'  T 0.01E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         VITE COMP 1
SCOU  3 'v_03'  T 0.02E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         VITE COMP 1
SCOU  4 'v_04'  T 0.03E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         VITE COMP 1
SCOU  5 'v_05'  T 0.04E-3 SAXE 1.0 'init_abscissa' INIT LECT xaxo TERM
                         VITE COMP 1
TRAC  1  2  3  4  5 AXES 1.0 'VELOC. [M/S]'
LIST  1  2  3  4  5 AXES 1.0 'VELOC. [M/S]'
RCOU 11 'v_01' FICH 've1d27a.pun' RENA 'v_01_27'
RCOU 12 'v_02' FICH 've1d27a.pun' RENA 'v_02_27'
RCOU 13 'v_03' FICH 've1d27a.pun' RENA 'v_03_27'
RCOU 14 'v_04' FICH 've1d27a.pun' RENA 'v_04_27'
RCOU 15 'v_05' FICH 've1d27a.pun' RENA 'v_05_27'
TRAC  1  2  3  4  5 11 12 13 14 15 AXES 1.0 'VELOC. [M/S]'
COLO NOIR NOIR NOIR NOIR NOIR
     ROUG ROUG ROUG ROUG ROUG
QUAL VITE COMP 1 LECT  6 TERM REFE  4.96403E-1 TOLE 1.E-2
     VITE COMP 1 LECT 16 TERM REFE  4.96376E-1 TOLE 1.E-2
*=============================================================
FIN

BEGIN DESCRIPTION

This test checks adaptivity in 1D (FUN2 element) with
INDI VITE plus ACCE on Verdugo's modified problem (1D pulse).

Three solutions are obtained:

1. VE1D27 : reference solution without adaptivity, using
   160 FUN2 elements
2. VE1D17 : adaptive solution with a base mesh of 20 FUN2 elements.
   The mesh is initially refined to level 4 bt INIT ADAP directive.
   Then adaptivity is piloted by INDI VITE.
   The solution presents oscillations due to intermittent
   mesh unrefinement/refinement as the curvature of the
   velocity profile passes through a zero value.
3. VE1D18 : adaptive solution with a base mesh of 20 FUN2 elements.
   The mesh is initially refined to level 4 bt INIT ADAP directive.
   Then adaptivity is piloted by INDI VITE ACCE.
   The solution is smooth and in excellent agreement with the
   reference. When the curvature of the velocity profile is zero
   that of the acceleration profile is not, and vice versa.

END DESCRIPTION
```

List of input files

Europe Direct is a service to help you find answers to your questions about the European Union
Freephone number (*): 00 800 6 7 8 9 10 11
(*) Certain mobile telephone operators do not allow access to 00 800 numbers or these calls may be billed.

A great deal of additional information on the European Union is available on the Internet.
It can be accessed through the Europa server http://europa.eu/.

How to obtain EU publications

Our priced publications are available from EU Bookshop (http://bookshop.europa.eu),
where you can place an order with the sales agent of your choice.

The Publications Office has a worldwide network of sales agents.
You can obtain their contact details by sending a fax to (352) 29 29-42758.

European Commission
EUR 26697 EN – Joint Research Centre – Institute for the Protection and Security of the Citizen

Title: Adaptivity in Shell/Beam/Bar Elements in EUROPLEXUS

Authors: Folco Casadei, Martin Larcher, Georgios Valsamos

Luxembourg: Publications Office of the European Union

2014 – 74 pp. – 21.0 x 29.7 cm

EUR – Scientific and Technical Research series – ISSN 1831-9424

ISBN 978-92-79-38670-1

doi:10.2788/85346

Abstract

The present work is a first attempt at extending mesh adaptivity to 3D shell elements. The Q4GS 4-node shell is taken as a first example because it is one of the most efficient and used shell elements in EPX. Then, also the T3GS triangle (which is similar to Q4GS as far as formulation is concerned) is considered. Finally, the case of 2D shells, i.e. of 2-node "segment-like" elements, is considered (this element shape can also represent beams or bars in either 2D or 3D). The treatment of shells/beams/bars presents some particularities with respect to the continuum elements considered in previous work.

LB-NA-26697-EN-N

JRC Mission

As the Commission's in-house science service, the Joint Research Centre's mission is to provide EU policies with independent, evidence-based scientific and technical support throughout the whole policy cycle.

Working in close cooperation with policy Directorates-General, the JRC addresses key societal challenges while stimulating innovation through developing new methods, tools and standards, and sharing its know-how with the Member States, the scientific community and international partners.

Serving society
Stimulating innovation
Supporting legislation

Publications Office

ISBN 978-92-79-38670-1
doi:10.2788/85346